

PET User Notes

Publication of the PET User Group Box 379 South Bound Brook, NJ 08880

VOLUME 2 ISSUE 1

PRICE \$1.50

TWO-HEADED EDITORIAL

Effective with this issue, the User Notes is operating with additional editorial and administrative help, which will enable us to keep to a more regular publication schedule.

Since we are starting our year so far behind, we will publish the Notes on roughly a six-week cycle, which will have us caught up by the end of the year.

The new help is in the form of Roy O'Brien, who has been active with the New Jersey Computer Club for the past three years. Roy will be responsible for the administrative end of things, including handling the mail and subscription activity, as well as picking up part of the editorial function.

Gene Beals will remain involved in the work of editing, and will also continue to operate the Tape Exchange Program.

In the past year, it has been very gratifying to see, in spite of the relatively meager documentation from Commodore, the emergence of sophisticated programs and routines for the PET. It's a remarkably potent little machine, and we're looking forward to further software development.

At the same time, each month we hear from new subscribers who have just bought a PET, and are in need of beginner-level information to get up-and-running.

To accommodate both the advanced and the neophyte PET owner, Gene and Roy are sharing the editorial work on the basis where Gene is mostly concerned with the advanced group, and Roy will see that the newcomers are not neglected. However, we work closely together, and mail may be sent to either one; the most important thing being that two people are (or ought to be) more effective than one.

We are strong believers in the need for an independent forum for PET owners to exchange information and ideas, and publish their work. Although we are more than pleased with the number of people who have written and submitted items for publication, we would like to repeat that all material is welcome...remember, regardless of where you are on the experience curve, there are others who can benefit from your work, and vice-versa.

Correspondence with individuals and PET groups is a vital part of this activity, and newsletter exchange is invited. If there is no group in your area, and you would like to start one, or contact other PET owners, send a postcard and we'll print an announcement. If you have any questions, let us know—it's a big help in trying to decide what to include in the Notes.

INDEX

New Commodore Products	
Beals	2
Cassette Exchange.....	3
Ultimate Graphics	
Covitz	4
Watching A Cassette Load	
Butterfield	7
Controlled Restore	
Wuchter	7
ROM Subroutines	
Zimmerman.....	11
Fast Graphics	
Swan.....	13
Single-Step Trace	
Russo.....	16
Hi-Resolution Plot	
Velders.....	18
Memory Test	
Velders.....	22

PET User Notes is published 6 times a year by the PET User Group, Box 379, South Bound Brook, New Jersey 08880.

Subscription is \$6 for 6 issues in U.S. and Canada, and \$12 for airmail to all other countries.

Co-Editors:

Gene Beals
Box 371, Montgomeryville, PA 18936
Roy O'Brien
Box 379, South Bound Brook, NJ 08880

Copyright © 1979, PET User Group

NEW COMMODORE PRODUCTS

A QUICK REVIEW

Gene Beals

As most of you know by now, many of the new PET products are being delivered. My understanding of current status is as follows:

PET 2001-16 is available in two models—one with a full-size keyboard (with cassette deck external and optional), and the other with the small keyboard and cassette built-in. Price is \$995.

PET 2001-32 is also available in the same two configurations at \$1295.

The PET 2040 Dual Disk is available in limited quantities at a price of \$1195. A model 2040A was just announced (July availability) as a single drive version with the same cabinet and controller board at \$895. Addition of a second drive will be possible for approximately \$400. The previously announced 2041 Drive has been cancelled (the unit was to attach through memory expansion connector rather than IEEE as 2040 does). While the 2040 models will operate with 8K machines as well, the new ROM set to allow the connection will not be available until about July (the ROMs are to be included in the price of the disk drive, however).

PET printers should be available in May (the 2023 Pressure Feed about mid-month at \$849, and the 2022 Tractor Feed at month-end for \$995). The 2021 Electrostatic Printer was cancelled.

Aside from the outward appearance of the full keyboards, the new models have a number of operational differences compared to the 8K unit. All the previous bugs and idiosyncrasies seem to be corrected, and additionally there are several enhancements.

Accesses (PEEK or POKE) to screen memory no longer generate snow.

RND(0) and RND(1) appear to be seeded by the timer, since values are random when first called after power-on.

When lower case is enabled, the shift operates as on a standard typewriter (shift provides upper case). Case shift on the screen however, is a function of the character ROM, so that when going to an alternate device (a printer), upper and lower case are reversed.

The signal which blanks the video on the 8K is not connected on the newer models, so POKE 59409,52 no longer works (ROM routines still use the address, however, causing a line scroll).

NMI Vector is initialized to BASIC "warm start".

When output is directed to an alternate device, the ASCII "space" code (\$20) is used for all BASIC-supplied forward spacing. This means that TAB, ";", and "," will provide proper spacing on your printer.

PEEK address range is no longer restricted.

Execution of some code seems to be slightly faster due to more efficient programming and better use of page 0.

Screen print is faster because extra code to maintain separate POS pointer was removed.

Cursor home, up, and down are now tracked properly by POS.

Machine Language Monitor is in ROM, and the BRK vector is initialized to point to Monitor (SYS 1024 gets you there). The MLM code is a little more compact, and Save and Load have revised syntax.

Operating system addresses for the new models have been changed extensively. Almost all locations in low memory (pages 0, 1, and 2) are different, with the exception of the cassette buffer areas. The new manuals which come with the PET 16 and 32 have a complete memory map for this area.

ROM routines, while essentially the same, have been reassembled to incorporate changes and corrections. Almost all segments are in slightly different locations. (If anyone needs a copy of the revised ROM addresses, send me a note. I have a list compiled by Jim Russo in which the new locations are listed in parallel with the old on a map previously identified by Jim Butterfield.)

I'm certain many people will be slightly upset by the address changes in both low memory working storage and the ROM area, since many programs made extensive use of these locations. If you programs have decent documentation, address changes should be relatively minor. If you purchased some of the "commercial" programs which are not documented or have deliberately unexplained code, you are at the mercy of the vendor.

A few programs will be difficult to convert regardless of documentation.

Standard PET BASIC programs should run with no changes (PEEK and POKE into areas other than screen memory are generally non-standard). If you're still concerned, remember the admonition of the large system people: use only standard language implementations (i.e. FORTRAN IV, ANSI COBOL, etc.) unless you want to be hardwired to a particular piece of hardware.

Several changes inside the new machines are apparent also. The revised main logic boards use dynamic RAM. The 16K uses 4108, while the 32K has 4116 (or equivalent) chips. Both boards are fully populated, and it doesn't look as though you can upgrade 16K to 32K with a chip change. Video RAM is static and uses the MOS 6224 (2114) part.

The memory port connectors are .1" on center "spectra-strip" style. Pinouts are described in the manual (which has already been updated and reprinted). Two new power connectors are included on board to provide power for "daughter boards".

The 2040 Dual Disk appears to be quite powerful and flexible. Usable capacity is 171,500 bytes per drive (343,000 bytes on line).

The controller incorporates a 6502 and a 6504, as well as 4K of static RAM (2114). The processors apparently watch over head, track, sector position, buffer management, and data transfer on the IEEE bus.

My use of the disk to date has been limited to moderately simple operation—SAVE, LOAD, and sequential data files. Random Access file capability is incorporated in the 2040 design without leaving BASIC (a sample program is included in the preliminary disk user manual).

One final note: Jim Butterfield suggests there be an "official" designator for new ROM machines, since existent PETs may also be upgraded with new ROM. This would make it easier to annotate listings for programs using any of the changed addresses.

Hope the review helps. Special thanks to Jim Russo for the stack of notes he sent, and for the new ROM address list. Again, if you need a copy of the Russo (new ROM)-Butterfield (old ROM) list, send a SASE and one boxtop from your 32K PET to Box 371, Montgomeryville, PA 18936.

PROGRAMS ON TAPE

Partial list follows -- send for complete list.

The "programs on tape" service and exchange functions as a low cost method for software distribution. The copying cost is \$1 per program. We can put up to 4 programs on a C-10 tape, or 12 on a C-30 tape. Please add \$1 per tape (either C-10 or C-30) to cover tape cost and postage within U.S. or Canada. If any written documentation is available, a copy will be included with the associated program.

At this time we are behind on getting programs copied onto tape and returned to you. We just re-acquired a person who indicated willingness to endure the drudgery of saving programs, so we'll be catching up in the near future.

If you have a program to contribute to the list (one which we don't have, or an updated or enhanced version of an existing program), please send it on tape. We will save it for the exchange and return a program of your choice. Again, due to space and time limitations, only a partial program list (primarily some of the newer programs) is included. Write us at PO Box 371, Montgomeryville, PA 18936 for requesting programs and for the complete list.

Among the new 2programs are several by Mark Zimmermann (Caltech 130-33, Pasadena, CA 91125). Some of these have been, or will be published in the various micro magazines. Mark granted permission to include the programs in the exchange -- but you might want to acquire the various publications for more complete documentation.

TINY ASSEMBLER -- Mark Zimmermann - Personal Computing 12/78
MAZE MODULE -- Mark Zimmermann - incomplete, but could be used as nucleus of game.
GAS -- Mark Zimmermann - machine language module for physics simulation to be published in BYTE.
GAS DRIVER -- Mark Zimmermann - BASIC driver for GAS.
BLACKBOX -- Mark Zimmermann - Parker Bros. game. Try to locate position of 5 balls in box by shooting probes into box.
INTERMOD -- Bill Stinson - IM calculation program for hams or communication techs.
ACTIVE FILTER -- Bill Stinson - Design program adapted from IBM library.
FREQUENCY COUNTER -- Bill Stinson - Basic 3 digit counter - will include copy of Bill's interface and connection to parallel port.
KLINGON CAPTURE -- Mark Turner - grid game from KILOBAUD
SIMON -- Gary Mayhuk - sound repetition game
OTHELLO/2 -- F. Dunlap. Modified by J Mendenhall
DIGIT SPAN -- number recall in sequence and reverse sequence for progressively long numbers
CENTRAL LIMIT -- Dave Heise - graphs results of repeated samples of any given size, showing averages tend to be normally distributed.
CHI SQUARE -- Dave Heise - constructs repeated random 4-fold tables and computes significance test for each. Nice graphics representation of statistical values
MONEY CHANGER -- Max Yoder - give correct change for amount shown. Basis of a good educational program. No graphics.
BABY -- Sally & Stan Klein - try to keep your sanity while feeding & diapering your baby yet have time to yourself
SAM -- R.Tansony - shoot down enemy planes from your missile base
CHECKERS -- William Anderson
PET ORGAN -- Wm. Anderson - Sound plays 3 octaves of notes
STARWAR -- John Ball - save planet from enemy fighter planes. 10 levels of difficulty.
LONG DIVISION -- E Lichten-long division tutorial by grade level
NEAT PROGRAM -- Chris Crawford
TANKS -- E.Lichten - shoot Russian tanks which are faster & able to move through the mine fields
FAIR OPTION VALUE -- Joe Kot - Evaluate stock option prices.
TRACE -- Brett Butler-traces basic program or direct entries
KEYBOARD -- Neil Harris/J Butterfield - musical keyboard, will remember and play back tune

MARKS -- J Butterfield - keeps track of names and grades, shows low, high, and average. Keeps tape file.
PORTFOLIO -- J Butterfield - keeps track of stocks. Buy sell, evaluate, and save on tape.
MERGE -- J Russo/H Chow - merge two or more program segments
M7171 -- J Russo/H Chow - high monitor with merge capability
IMAGE -- P Rowe, modified by H Chow - save any portion of memory on cassette
TIME/COORD -- Adolf Futerweit - for sky watchers, provides the necessary calculations to prepare for a night of observing or program planning. Uses 2 cassettes. NOTE: this has 2 programs & extensive documentation and will cost \$3 instead of \$1.
DEBUG -- E Blaschke - poses a programming problem regarding floating point calculation, asks you to debug, and provides solution (if you give up).
MT6671 -- Jim Russo - excellent version of high monitor and disassembler with single step capability.
ASSEMBLER -- Mark Zimmermann - quick assembler with tape object code save and load capability.
BLACKBOX -- Steve Michel modified from Creative Computing - guess location of atoms in space
QUARTIC -- W.Hawes - solves quartic, cubic, and quadratic equations.
BAIRSTOW -- W. Hawes - solves Nth order polynomial equations.
COMPLINEQ -- W. Hawes - solves linear equations with complex coefficients. Uses a modified Gauss-Jordan elimination method.
TANK WARS -- Ralph Bressler - 2 player tank fight game.
EXAGON -- Mac Ogelsby adapted by James Parsly - puzzle game for you to learn by playing. You against the Computer.
SAND PILE -- Mac Ogelsby adapted by J.Parsly - try to get a sand pile level without losing any sand.
LINE MAZE -- Frank Tyniw - adaptation of Andy Fraley's short maze with sound. An obstacle course one player game.
AFO -- Japanese program with phenomenal graphics and animation.
TOKER -- anonymous Michigan User (of PET) - excellent graphics. This program will be an underground classic.
SOUND DEMO -- CAP Electronics - program is an advertisement but still a nice demo for CB2 sound.
ARROW -- J Butterfield - in Volume 1, Issue 7
ELEMENT DRILL -- Richard Larson - intro for beginning chemistry students on element abbreviations
IONIC DRILL -- Richard Larson - for beginning chemistry students on chemical symbols and ion charge.
FILE & SORT -- enhanced version of filing and record retrieval system.
BASE -- F Campbell - convert from one base to another
STICKS ON HILL -- super graphics demo
OPTICAL ILLUSION -- super graphics demo
1000 MILES -- F Covitz - computer version of Mille Borne. It's very addictive.
ORDINARY DIFFERENTIAL EQUATIONS-1st & 2nd ORDER -- Karlheinz Lehner - computes approximate ordinates to the solution curve given boundary values by a Runge-Kutta method
KNIGHT'S TOUR -- F Chambers - solution to classical chess puzzle, PET graphics and animated gimmicks
PETALS AROUND THE ROSE -- Wm Rothanburg - clever puzzle
FRACTION -- Don Johnson - educational game to teach fractions
TI FIGHTER -- with sound
TAPESTRY -- Frank Chambers - pattern generator using a couple of the symbols hidden in lower case.
PINBALL -- Bruce Jaeger - good graphics and sound.
DISASSEMBLER -- Karlheinz Lehner - Disassembles machine code in KIM (MOS Technology) assembly language.
CALENDAR -- Ernest Blaschke - draws a calendar of any month & year from 1904-2099.
UNIT CONVERSION -- Ernest Blaschke - converts weight, distances, time, pressure, temperature (metric or imperial)

Letter from Frank Covitz

Dear Gene,

On January 21, 1979 at approximately 3 pm, I witnessed what may have been a unique event. For the first time, I believe, I saw a demonstration of the capability of controlling individual pixels on the PET screen!!

As you know, the PET screen normally allows a display of 1000 characters, each within an 8 x 8 pixel matrix. This comes out to a display format of 320 horizontal by 200 vertical pixels, or a total of 64,000 pixels. However, because of the way the operating system manages the video (1K bytes of screen RAM plus character generator ROM), the user can only access by characters available on the keyboard. I have often thought how desirable it would be to have control over each of the 64,000 pixels. I can now state that it can be done!!

The key is an elegant piece of hardware called "visible memory" (VM) designed by Hal Chamberlin and Dave Cox of *Micro Technology Unlimited (MTU)* - the same people who gave us the 6502 4-part harmony music software and DAC/filter/audio amplifier board. The VM is essentially an 8K dynamic RAM board but with the following feature - During the phase 2 part of the clock cycle (when the 6502 does not use the address bus) the dynamic RAMS are refreshed and at the same time a video signal is generated. To the 6502, the system behaves like 8K bytes of RAM. However, the video signal (when displayed on a monitor) allows the user to see every bit of every byte. Unlike the PET internal video, the VM does not have any "snow" on it during read or write operations.

The VM, at present, needs an interface card to get the proper addressing and to permit the use of the PET video monitor with VM. The January 21, 1979 date mentioned above was the day I brought my PET to MTU headquarters in Manchester, NH. After Hal Chamberlin got a look at the PET circuit board with a 'scope, he was quickly able to put together an interface breadboard which compatibilized the VM to the PET. The VM can be turned on via a hardware switch or, better yet, under control of an unused I/O port. This allows software to select either VM or PET video. The people at MTU are at present designing a finalized board for interfacing PET with VM. They may very well have this completed by the time this letter gets out, so I would definitely write them for info and pricing. Their address is:

Micro Technology Unlimited
P.O. Box 4596
841 Galaxie Way
Manchester, NH 03108

With care, the VM and interface can be fitted *inside* the PET.

MTU's hardware and supporting software is, as far as I am concerned, superb. Although designed mainly for KIM-1, their products should run on most 6502 based products, i.e., the PET. If their PET market grows, they will undoubtedly consider PET-specific hardware and software.

With a little bit of programming effort, I have already converted Hal Chamberlin's VM *Life* software to run on the PET, including the clever demo where the word "Life" is drawn on the screen and the system then applies the *Life* algorithm to it. A winking cursor, cursor control keys (including CLR), pause, and return to BASIC are supported in the PET modified version. Each generation takes about 4 to 6 seconds to complete, super fast when you consider that 64,000 bits are involved!!

With the PET being a video oriented machine, I predict that VM will open up a whole frontier in software - games with true animation, high resolution graphics, special character fonts, etc. Please spread the word.

F. Covitz

Ed. Note: The Visible Memory and interface Frank describes were exhibited by MTU at the Trenton Computer Festival and really drew a crowd, including some surprised and curious Apple owners. With this add-on, you can run high resolution graphics or return to the regular PET character set and have an additional 8K of RAM available for programs. MTU's ad on p. 23 gives an idea of what this display can do.

—O'B

Now... Inexpensive.. hard copy from Commodore PET®

*THE SIGMA PI-1 INTERFACE COUPLES
A STANDARD PET TO THE SOUTHWEST
PR-40 PRINTER VIA THE IEEE PORT

*NO ADDITIONAL CABLE OR PLUGS ARE
REQUIRED IF THE PRINTER HAS BEEN
BUILT TO PLUG INTO THE SOUTHWEST
6800 COMPUTER SYSTEM

\$39⁹⁵

*FULLY ASSEMBLED

*POSTPAID

*SATISFACTION
GUARANTEED

SIGMA ASSOCIATES

BOX 2065

PRINCETON NJ 08540

FULL LINE OF PET PRODUCTS—INTRODUCTORY SPECIALS

PET 8K	\$695
PET 16K (16N)	\$860
PET 32K (32N)	\$1125

The 16K N and 32K N have a full size keyboard and numeric pad with graphics on keys. Additional sockets provided on board for 8K ROM expansion. Machine language monitor in ROM.

Please add UPS shipping and insurance on PET products

PET EXTERNAL CASSETTE	\$83
------------------------------	------

Use as a second cassette for any PET model, or as primary cassette for 16K or 32K models

PET Dual Disk Drive (2040)	\$1125
-----------------------------------	--------

Total of 343K bytes on two standard 5.25" disks. For use with any PET. Attaches to IEEE Port.

SOFTWARE FOR PET

TUNNEL VISION and KAT & MOUSE	\$7.95
--	--------

Two excellent maze programs. In Tunnel Vision, you view the maze from inside in perspective. In Kat & Mouse, you must find your way through the maze before the hungry KAT finds you. Each maze has only one solution, and each is unique. By Michael Riley.

KITE FIGHT & SIMON	\$7.95
-------------------------------	--------

Kites is a completely original 2 player action game with its own unique strategies. Includes 4 pages on real Indian Fighter Kites. By Michael Riley.

TANK BATTLE in machine language	\$6.00
--	--------

Provides a random battlefield with trees, lake, walls, and 6 to 8 character tanks. Laser cannons can fire over lakes, bounce off walls, destroy trees, and explode tanks. By Michael Riley.

TRAP (SOUND) in machine language	\$5.00
---	--------

Build a wall around your opponent and force him to crash. OPTIONS: Diagonal walls, wrap around, and invisible walls. By Michael Riley.

CMC WORD PROCESSOR	\$24.95
---------------------------	---------

The best available word processing software for the PET.

MICROCHESS by Peter Jennings	\$17.95
-------------------------------------	---------

Offers 8 levels of play to suit everyone from beginner to the serious player. It examines as many as 6 moves ahead.

BRIDGE CHALLENGER	\$13.45
--------------------------	---------

You and the dummy play 4 person Contact Bridge against the PET. The program deals hands according to your criterion for high card points. You can review tricks, swap sides, or replay cards after each hand.

STIMULATING SITUATIONS	\$13.45
-------------------------------	---------

Dr. C.W. Engels book of 10 interesting simulation games with complete documentation, instructions, and suggestions for modification and enhancement. All programs on cassette.

GRAPHICS UTILITY PACKAGE	\$13.45
---------------------------------	---------

ABTAPE1	\$8.00
----------------	--------

Includes Life, Biorythm, Othello, Mastermind II Multiprimer (math tutorial-deduction game), and Capture.

ADVENTURELAND	\$7.95
----------------------	--------

This is not the average computer game that you master within the hour. In Adventureland you wander thru various 'rooms' manipulating objects to find 'treasures' using 2 word commands. By Jeff Jessee and Scott Adams. REQUIRES 24K MEMORY.

BOOKS & ACCESSORIES

BASIC for Home Computers —Albrecht, Finkel, Brown. New programmed instruction book on MICROSOFT BASIC	\$5.95
--	--------

Hands-On BASIC with a PET —Peckham	\$10.95
---	---------

Beginners book with easy to follow, step by step approach to programming.

Stimulating Simulations —Dr. C. W. Engel	\$4.00
---	--------

6500 PROGRAMMING MANUAL MOS Technology	\$6.50
--	--------

6500 HARDWARE MANUAL MOS Technology	\$6.50
---	--------

HEWLETT-PACKARD IEEE—488 STANDARDS Manual	\$1.50
--	--------

PROGRAMMING A MICROCOMPUTER: 6502 Caxton Foster	\$8.95
---	--------

PET IEEE to RS-232 or 20 ma printer interface by GPA—Assembled and tested. Selectable baud rates 110-1200. Connectors included	\$59.50
---	---------

PROTECT-A-PET Custom fitted clear vinyl dust cover with quality stitched seams for those rare occasions when you're away from your PET	\$9.95
--	--------

KIM-1 Includes manuals and documentation	\$161
---	-------

SYM-1 Synertek Systems—expanded 6502 single board system	\$238
---	-------

Prices include UPS shipping (except where noted) for prepaid orders over \$10

CASSETTE TAPES Premium quality high output, low noise in 5 screw housings with labels. Boxes separate.	
C-10 (5 min/side)	10/\$5.95 50/\$26 100/\$48
C-30 (15 min/side)	10/\$7.00 50/\$31 100/\$57
ALL TAPES GUARANTEED Soft plastic boxes	10/\$1.00
Norelco-style hard plastic boxes	10/\$1.25

PET Edge CONNECTORS	
Second Cassette Port	\$1.60
User or IEEE—488 Port	\$2.25
Memory Port—solder eyelet	\$4.45
solder tail	\$2.95

VERBATIM 5.25" diskettes for PET	10/\$28.50
---	------------

Godbout S-100 8K Static RAM —assembled	\$139
---	-------

S-100 16K Static RAM (Problem Solver Systems)	\$279
--	-------

BETSI PET to S-100 Interface Motherboard	\$149
---	-------

Four slot motherboard—On board sockets and decoding circuitry hardware with four 100 pin edge connectors. Assembled.

BETSI Kit —one S-100 connector	\$119
---------------------------------------	-------

Power Supply for BETSI and static RAM boards—8 volt/6 amp. Assembled in case.	\$34
---	------

HEATH H14 TRACTOR FEED PRINTER	\$745
---------------------------------------	-------

80, 96, and 132 characters per line, 6 and 8 lines per inch, baud rates to 4800 with multi-line buffering.

A B COMPUTERS

115 E. Stump Road, Montgomeryville, PA 18936 (215) 699-8386

SAWYER SOFTWARE

Business Software for your 8K PET™

★ ★ ★ NEW ★ ★ ★

ACCOUNTING PACK 1 —

Accounting Pack 1 is a general ledger package designed for small businesses and homeowners. It contains check journal, general ledger, income statement (current ytd, previous month ytd and current month), balance sheet (current month and previous month). There are 15 commands and 6 reports that can be generated. The system uses a unique single-entry bookkeeping system and can hold up to 50 entries per period (month, week, day) and up to 40 different accounts. Each period's data is kept on convenient cassette tapes. Utilizing the general ledger command the user can view the general ledger entries for the month from Assets to Expenses or stop in midstream and view one particular account. Or the user can type in an account name such as "Advertising" and view the entries for that month. The Accounting Pack 1 program includes a checkbook reconciliation routine which aids in finding checkbook errors. Sawyer Software plans updates to Accounting Pack 1 to enable the user to use a printer, floppy disk or more memory. Accounting Pack 1 is well documented, with a newly updated User's Manual. Several businessmen are using Accounting Pack 1 and have written to us their satisfaction with the program along with their purchase of other software.

\$25.00

SCHEDULE PLANNER —

Schedule Planner can be used by secretaries, receptionists, housewives or anyone wanting to plan and have at their fingertips their own schedule. Data entered is date, time, priority and description. The commands allow the schedule to be shown for a particular day, request of time or the "viewing" of appointments according to importance.

\$15.00

SCHEDULE PLANNER #2 —

Schedule Planner #2 includes all the features of Schedule Planner, but is used for one or more individuals. Utilizing Schedule Planner #2 a customer can call in asking when his appointment with Dr. Jones is and in seconds the receptionist can give the date and time. Or Dr. Jones can find out his schedule for the day. With the viewing command, an appointment at 12:00 on a particular day will display on the screen at that time allowing receptionists and secretaries to validate appointments.

\$20.00

COMING SOON —

Business Software for your 16K TRS-80. Write or call for details and availability.

ACCOUNTING PACK II —

Accounting Pack II is a much more powerful version of Accounting Pack I. It has all the features of Accounting Pack I, plus up to 250 entries per period, MENU, Optional debit-credit of entries, easy addition or deletion of accounts, formatter for reports, intelligent report generator and single entry for sales transactions. Accounting Pack II requires at least 8K additional memory for your PET. OPTIONS: 1. Accounting Pack I to Accounting Pack II data file converter (free to previous owners of Accounting Pack I).

\$45.00

BUSINESS ANALYSIS —

Business Analysis allows management to have available to them information for financial planning decisions. Up to 4 years of balance sheet and income statement information can be entered with resultant analysis in the areas of liquidity, leverage, profitability and activity. The ratios generated for each year are the: current ratio, acid test, debt-nw, profit-nw, profit margin, sales-rec, sales-inv, sales-wc, with a brief explanation of each. The growth analysis gives the yearly growth in 5 areas and the average growth in 5 areas. Future growth analysis projects figures for the next year.

\$30.00

FINANCIAL PACK I —

Financial Pack I includes Amortization Schedule, Calendar (which figures days between two dates), Rebate (which is calculated on the Rule of 78's), Compound, Annuities and Monthly payment.

\$15.00

CHESSBOARD —

Chessboard allows two players to play chess on your own PET graphic chessboard. This program displays each player's time and records each move to allow you to playback the game at another time.

\$15.00

PAYROLL —

Especially designed with the small businessman in mind. Utilizing cassettes can record data for any number of employees (8 employees per cassette). Computes tax information and updates totals for quarterly and yearly reports. Employees can be salaried or hourly and pay periods can be either weekly, bi-weekly, semi-monthly or monthly.

\$30.00

BUSINESS GRAPHIC PACK 1 —

Business Graphic Pack 1 is a simple program to use, but professional in output. The graph includes title, labeling of axis, dual graphic ability, whether the data is in Mill's, 100's or 1000's, and an optional x-axis = date and labeling of the x-axis with month and year. Entry is as easy as typing the title, # of entries, the X,Y value (Jan. 15, 1978 would be entered as 115.78), entering if the x-axis = date, if the user wants crosshatching and then graphing. The program also includes Nth order and Geometric regression to give the user a formula for his set of data (if possible).

\$25.00

CHECKBOOK RECONCILIATION —

Designed for ease of use and to find those troublesome checkbook errors, this program balances your checkbook and bank statement from month to month. It locates over ten different types of errors and instructs the user on correcting them, including bank statement errors made by your bank. Records data on cassette for next month's balancing - saving re-entry of figures.

\$25.00

CALCULATOR —

Especially for the accountant-bookkeeper. This program not only allows your PET to function as a calculator, but also has such features as Lister (which verifies and displays errors between two tape listings) and Matrix (which adds columns vertically and horizontally).

\$10.00

All Programs include documentation, are in BASIC and on cassette.

*PET is a trademark of Commodore Business Machines, Inc.

Dealer inquiries invited.

Sawyer Software

201 Worley Road
Dexter, Mo. 63841

314-624-7611



ANNOUNCING

THE
commodore
PET

personal computer



Now Available in the Washington D.C. Area From:

The
COMPUTER SYSTEMS STORE

TYSONS CORNER VA. NEAR HOLIDAY INN
1984 Chain Bridge Rd., McLean, VA 22101
703 - 821-8333

"A FULL LINE COMPUTER STORE: SALES - SERVICE - SOFTWARE"

WATCHING A CASSETTE LOAD

Jim Butterfield

It may not be too useful, but it's very satisfying to watch a program coming in from cassette tape. Much of what comes in will look like gibberish, since the program contains obscure things like pointers, flags, and tokens. But it's interesting to see, and here's how you can do it.

Step 1: Load any Basic program on cassette 1. The program doesn't matter; the LOAD activity sets up certain internal things that will help us.

Step 2: Set up the cassette with any Basic program ready to load. A short one would be good; that way you may catch the whole program on the screen. But any Basic program will do.

Step 3: Set graphic mode with POKE 59468,14. This may help you spot a few recognizable pieces of your program.

Step 4: Give SYS 62894. PET will ask you to press PLAY. Do so, and in twenty seconds or so, PET will report FOUND...and stop.

Step 5: Clear the screen so you'll get a better view of the program as it comes in. Now move the cursor down to a few lines from the bottom of the screen.

Step 6: Enter POKE 636,128:POKE 638,132:SYS 62403

Step 7: Sit back and watch the program load to the screen. You won't be able to run it, of course, since it's in the wrong part of memory...but isn't it fascinating to watch?

CONTROLLED RESTORE

Earl Wuchter

This routine will reset the DATA pointer to the beginning of any DATA line. It is useful when some data statements are to be read more than once from different places in a BASIC program. To use the routine, set RS= the desired DATA line number, and GOSUB 6000. A word of caution: since the line number is a BASIC constant, it will not be automatically changed if the program is resequenced. The sample program shows how the sub-routine can be used.

```
10 REM SAMPLE PGM USING CONTROLLED RESTORE
20 READ A,B,C,D,E :PRINT A;B;C;D;E
30 DATA 1,2,3,4,5
40 DATA 41,42,43,44,45
50 READ F,G,H,I,J :PRINT F;G;H;I;J
60 RS=40 :GOSUB 6000
70 READ F,G,H,I,J :PRINT F;G;H;I;J
80 END
6000 IF PEEK(142)+16*PEEK(143)>=RS THEN RESTORE
6010 FOR J=0 TO 5 :SV%(J)=PEEK(J+142) :NEXT
6020 READ J$
6030 IF PEEK(142)+16*PEEK(143)<RS GOTO 6010
6040 FOR J=0 TO 5 :POKE 142+J,SV%(J) :NEXT
6050 RETURN
```

Output: 1 2 3 4 5
41 42 43 44 45
41 42 43 44 45

PET 2001 • PET 2001 • PET 2001 • PET 2001 • PET 2001 • PET 2001 • PET 2001

Programmer's Guides for the PET™



TIS

WORKBOOKS FOR THE COMMODORE PET 2001
Getting Started with Your PET WB-1 \$4.00

Covers the fundamentals of PET BASIC: calculator and program mode, data input and output, data representation, program storage on the cassette.

PET String and Array Handling WB-2 \$3.95

Covers string and substring search, concatenation, replacement and manipulation.

PET Graphics WB-3 \$4.95

Covers use of cursor control and special graphics symbols to draw plots, histograms, and sketches.

PET Cassette I/O WB-4 \$4.95

Covers OPEN, CLOSE, string and numeric data files.

Miscellaneous PET Features WB-5 \$3.95

Covers the clock, random number generator, upper and lowercase alphabetic characters, saving memory space, etc.

PET Control and Logic WB-6 \$3.95

Covers IF, GOSUB, logical operations, and ON X.

Add \$1.50 for shipping and handling.

Dealer
Inquiries
Invited
TIS
P.O. Box 921
Los Alamos, NM 87544
Money back guarantee

We also sell PET Software. Write for details.

PET is a trademark of Commodore Business Machines

PET 2001 • PET 2001 • PET 2001 • PET 2001 • PET 2001 • PET 2001 • PET 2001

PET 2001-8 Computer--Standard PET with integral cassette and calculator type keyboard. 8K bytes of memory. (7167 net). \$ 795.00
 PET 2001-16N Computer--PET with 16K bytes of memory and large keyboard with separate numeric pad & graphics on keys. External cassette optional. (15,359 net) \$ 995.00
 PET 2001-16B Computer--As above but has standard typewriter keyboard. No graphic keys. \$ 995.00
 PET 2001-32N Computer--Identical to 2001-16N with 32K bytes of memory. (31,743 net). \$1,195.00
 PET 2001-32B Computer--Identical to 2001-16B with 32K bytes of memory. (31,743 net). \$1,195.00

PERIPHERALS

PET 2021 Printer--80 column dot matrix electro-static printer with full PET graphics capability. \$ 549.00
 PET 2022 Printer--80 column dot matrix printer with plain paper or forms handling tractor feed. Has full PET graphics. \$ 995.00
 PET 2023 Printer--80 column dot matrix printer. Plain paper printer with full PET graphics. \$ 849.00
 PET 2040 Dual Drive Mini Floppy Disk--Dual drive intelligent mini floppy system. 343K net user storage capacity. \$1,095.00
 PET 2041 Single Drive Floppy Disk--Single drive intelligent mini floppy. 171.5K user storage. \$ 595.00
 PET External Cassette-Cassette player/recorder to use with PET 2001/8/16/32. \$ 95.00
 PET User Manual--160 page expanded user manual. \$ 9.95
 MICROTRONICS__M-65 Ham Interface--Wired \$ 99.95
 --Kit \$ 69.95

SOFT-WARE

V-1--CANNON BALL--Requires 2 players. Each calculates the angle and velocity of a cannon ball dependent upon wind and terrain conditions (which change each game).
3D TIC TAC TOE--Requires 1 player vs. the PET. The board consists of 4 layers of 4 x 4 boards. BOTH \$ 5.95
 V-2--WEATHER WAR--Requires 2 players taking alternate turns directing various types of storms against their neighbor's house. Correct calculation of storm's angle (and type) destroys a house in part.
BINGO--For 1 to 4 players. Four different Bingo cards are put on the screen and the PET calls out the numbers. BOTH \$ 5.95
 V-3--MISSILE RANGE--Requires 2 players, each player taking alternate turns firing a missile at a group of targets. Angle of missile is under players control.
AWARI--Requires only 1 player playing against the PET. Awari is an ancient African game in which the player and the PET move a group of stones around a board consisting of 6 bins on each side. BOTH \$ 5.95
 V-4--SPIN TO WIN--Requires 1 to 9 players and the PET (if you want it to). Each player takes a turn at stopping the spinning roulette-type wheel.
BATTLESHIPS--A famous naval game requiring 1 or 2 players. Each player places his/her naval fleet on the board while other player looks away. BOTH \$ 5.95
 V-5--DOG RACE--Requires 1 to 9 players. A game consists of players placing bets on 5 different dogs that race two laps around the track.
MASTERMIND--Requires 1 player. PET selects a number and you try to guess correct digits and their positions. BOTH \$ 5.95
 V-6--HOMEWORK--Lets your child practice against the PET. Addition, subtraction, multiplication, and division--all ran on one program. For students who need practice on higher numbers. \$ 5.95

ALL STOCK ITEMS ARE PREPAID IN LOWER 48. ALL ITEMS U.S. FUNDS.

VARITRON
 Box 729
 Chinook, MT 59523

ANNOUNCEMENTS

So that everyone knows my biases, I work full time at A B Computers in Montgomery, PA. A B Computers sells PET computers among other things. Since the PET User Notes were not commercially oriented, I have made every effort to keep my two activities separate. I hope I haven't disappointed anyone, and that I have been moderately objective during the past year. My apologies for not having been as responsive or as timely as I should have been with the User Notes. Thanks to everyone for the support extended. I'm sure the Notes will be more effective with Roy involved.

—G.B.

Don Henderson, Box 604, Westminster, CA 92683 announced an adaptor to allow a 2114 to be used in place of a 6550 RAM. The adaptor works for video RAM locations only (switch your video chips to bad 8K locations and use the adaptor) unless you build a decoder (Don has a PC board for this also). Price assembled is \$6.20 including postage.

ASM/TED, Macro Assembler and Test Editor for PET, Apple II, or SYM was announced by C.W. Moser, 3239 Linda Drive, Winston-Salem, NC 27106. The machine language package occupies 8K of memory starting at 2000 hex.

Dave Gomberg, 7 Gateview Court, San Francisco, CA 94116 is offering a PILOT package containing language processor, editor, simple program, teachers manual, and reference card for \$12. Also available for \$8 is a PILOT tutorial.

Chris Crawford, Pleiades Game Co., 202 Faro Avenue, Davis, CA 95616 has developed TANKTICS, a sophisticated simulation of World War II armored combat (you command a force against the PET). The package comes with instruction manuals, machine language and BASIC program cassette, battlefield map and playing counter, and costs \$15.

LAS VEGAS PET USERS are looking for members. Information can be had by contacting: Home Computers, 1775 E. Tropicana, Las Vegas, NV (702) 736-6363.

To all PET owners in and around Salt Lake City, UT interested in forming a PET Users Group contact Dee Husted, 4634 Eskesen Dr., Granger, UT 84120 966-0879.

Richard T. Tabbert, Glendale, AZ, would like contact with other PET users in Phoenix area. Contact him at 5014 W. Puget Ave., Glendale, AZ (602) 931-4544.

Northern Virginia PET User Group now called CAPE (Capital Area PET Enthusiasts); meeting 2nd Saturday each month at Patrick Henry Library, 101 Maple, Vienna VA. Contact Bob Karpen, 2054 Eakins Ct., Reston, Virginia 22091 860-9116.

FLOPPY DISK FOR PET! - S100 EXPANSION

ALL IN ONE

PETDISK SYSTEM from CGRS provides both a FLOPPY DISK and an S100 ADAPTOR for the PET! The Floppy Disk allows the Pet to instantly load and save programs and information. The S100 Adaptor allows the Pet owner to use the extensive variety of S100 peripheral boards: Memory, I/O, Voice Generators, Analog I/O, Printers, and even Telephone Interface Cards.

PETDISK uses the standard IBM 3740 format. (read disks from other systems I)



*PET is a trademark of Commodore.

PETDISK: SYSTEM

PACKAGE 1 \$ 799.95

MINIFLOPPY DISK DRIVE - 80 kb
EXS100 DISK CONTROL BOARD
DISK HOUSING- POWER SUPPLY
KM³ - WILSERV SOFTWARE PACK

Option 1 - Built in S100 mainframe: \$95.00

Option 2 - Software on EPROM: \$60.00

EXS100: DISK CONTROL BOARD

\$ 299.95

S100 size p.c. board, assembled and tested.
Connects to the PET memory expansion port.
Provides a Floppy Disk Controller and
Eprom socket, along with room for a
complete S100 Adaptor.

KM³: SOFTWARE

From WILSERV INDUSTRIES: Allows you to
Save, Run, Load, and Update programs from
the disk. Disk Commands, executed directly
or from a program, use existing BASIC format
preceded by a special character string.

CGRS MICROTECH

P.O. Box 368 SOUTHAMPTON, PA.

(215) 757-0284

18966

DEALER INQUIRIES WELCOME

The COMPUTER FACTORY

TO ORDER CALL TOLL FREE 800-223-7318



COMMODORE "PET"

only \$795 ^{USED} \$549

- 14K ROM Operating system
- 8K RAM Memory
- 9 Video Monitor
- Built in Keyboard
- Digitally controlled tape

Today's best value in personal computers, along with the latest in peripherals, are always in stock at the COMPUTER FACTORY

NEW! PET BUSINESS SYSTEM

The PET is now a truly sophisticated Business System with the Floppy Disk and Printer which makes an ideal cost efficient business system for most professional and specialized fields: medicine, law, research, engineering, education, etc.

PET 2001 -
16K \$995
32K \$1295

- 16 or 32K Bytes Dynamic RAM • 14K ROM Operating System
- Upper/Lower Case and Graphics
- Full Sized Business Keyboard
- Full Screen Editing
- Operating system will support multiple Languages (BASIC resident)
- Machine Language Monitor
- 8K ROM Expansion Sockets
- 9" CRT



PERIPHERALS FOR PET

- 24K Memory Expansion \$499
- 16K Memory Expansion 399
- PET to RS232 Serial 169
- 2 Way Serial Communication 219
- Modern Board for PET 375
- Analog to Digital Board 275
- For 16 Devices 95
- Second Cassette Drive 199
- Parallel Printer Interface 199



PET PRINTERS
2023 \$849
2022 \$995
(Includes Tractors)

- Bottom and Rear Tractor Feed
- 8 1/2" Paper Width
- 6504 Microprocessor Controlled
- 150 cps Bi-directional
- 4K ROM • 1K RAM
- Upper/Lower Case and Graphics
- 7 x 6 Dot Matrix



PET MUSIC BOX
Add music and sound effects to your programs. Compose, play, and hear music on your pet. Completely self-contained (no wiring). Free 3 programs including Star Wars theme, sound effects, etc. \$39.

DUAL DRIVE FLOPPY DISK 2040 \$1295

- 360 K Bytes Storage
- High Speed Data Transfer
- Plugs into IEEE Port
- 6504 Microprocessor
- 8K ROM Operating System
- 8K ROM Encoding and Decoding
- 4K RAM
- Uses Single or Dual sided Diskettes

SINGLE DRIVE FLOPPY UNIT 2041 \$895

- Intelligent mini floppy
- 171 5K net user storage

SALE

Come In Or Call For Special Low Prices
WE'RE MOVING TO LARGER QUARTERS AND MUST SELL EVERYTHING IN OUR INVENTORY

FREE \$35 of Software with purchase of any computer on this page



\$1495 Complete!
16K Model add \$200
32K Model add \$500

CompuColor II

COMPUCOLOR II Disk-Based Model 3 Advanced hardware and software technology gives you:

- 13" Color Display
- Advanced Color Graphics
- 51K Disk Built-In
- 16K ROM Operating System
- 8K RAM User Memory
- 4K RAM Refresh
- 8080A Microcomputer
- RS-232 I/O

Every unit comes with an extended DISK BASIC that has full file management capability resident in the COMPUCOLOR II in 16K of ROM. Color is fantastic, but COMPUCOLOR II has the power to handle complex tasks and small business applications. An impressive software library supplements your own creativity.

BUSINESS COMPUTER IMSAI

The low cost solution for all small business problems. A wide variety of software is available for all your needs. PCS series include dual floppies, 32K RAM, I/O, DOS, BASIC.

- PCS-42 (400KB) \$3295
- PCS-44 (780KB) \$3995

VDP-42 series adds video terminal, keyboard and VIO to above.

- VDP-42 \$4995
- VDP-44 \$5595
- VDP-80 \$7995



- EXIDY
 - 280 Processor
 - Keyboard and Numeric Pad
 - 8K RAM (up to 32K)
 - Serial and Parallel I/O
 - Dual Cassette I/O
 - 30 x 64 Display
 - 5100 Compatible
 - 64 User Defined Characters
 - 240 x 512 Resolution Graphics
- Modular design allows use of both cassettes for programs and ROM PAC Cartridges for future language introductions like: APL, COBOL, PILOT, FORTRAN, etc. Add monitor and tape or disk for complete computer system.

SORCERER

SPECIAL 12" Video Monitor for SORCERER (\$299 value) ONLY

\$125 with 8K unit
95 with 16K unit
65 with 32K unit

RADIO SHACK • PET • SORCERER • APPLE • COMPUCOLOR • ETC. PRINTERS • PRINTERS • PRINTERS

The COMPUTER FACTORY'S extensive inventory and wide selection of computer AXIOM (Parallel) 445 printers assures you of finding the printer AXIOM (Serial) 520 TRENDCOM 375 INTEGRAL DATA 795 well with all known personal computers. QUME or DIABLO 3400

Master charge Min Credit Card Order \$75 VISA Open Mon.-Fri. 10-6 Sat. 10-4

N.Y. residents add 8% sales tax • Same day shipment on prepaid and credit card orders • Add \$5 shipping for computers. \$3 for boards. \$25 each cassette tape.



only \$4145

BILLINGS MICROSYSTEM EXPANDABLE TO 2 MEGABYTES NEW!

- features: • 12" CRT • 64K RAM • Z-80 • Dual mini floppy
- 320 K Storage • Graphics • 94 Keys • 2 RS-232 ports • 1 parallel port

BUSINESS SOFTWARE AVAILABLE!



- APPLE II: APPLE is a completely contained computer system with BASIC in ROM, full ASCII keyboard in a lightweight molded carrying case. Expandable to 48K.
- APPLE II Disk \$595
- APPLESOFT in ROM \$204
- Interface Cards \$179
- MICROPRODUCTS: • EPROM Pgm. \$99.95
- EPROM Sock. Adpt. \$14.95
- APPLE II Modem \$129.95

SUPER SOFTWARE

Word Processing For PET This super advanced full function program will allow you to create text from PET or terminal keyboard. • INSERT • DELETE • CENTER • UNDERLINE • FULL SCREEN EDITING • MOVE LINES OR BLOCKS • SAVE TEXT ON TAPE • AUTO PRINT FEATURE...\$45

Apple's Moving Average Plot Program This fantastic program disk allows the statistician, mathematician, trader in stocks, money or commodities, the ability to maintain 30 database series of up to 300 values and plot 3 different moving averages of a series at the same time, in 3 different colors. Files can be updated, deleted, changed, extended, etc. A sure value disk at only \$40! Word Processing For Apple on disk...\$50

ANDERSON JACOBSON



- 841 I/O Terminal Ideal for word processing and small business: • ASC II Code • 15 cps Printout • High Quality Selective Printing • Use Keyboard for PET • Reliable heavy duty Mechanism • Completely Refurbished by A.J. • Service in 15 Major Cities
- Parallel \$1095
- Serial \$1195



INTERTEC's INTERTUBE video terminal Full screen editing 24 x 80 display and microprocessor controlled. Only \$870

SPECIAL SALE Buy any computer and take 10% off price of printer of your choice

TO ORDER CALL TOLL FREE 800-223-7318
The COMPUTER FACTORY 485 Lexington Avenue 750 Third Avenue New York, N.Y. 10017
(212) 687-5001 (212) PET-2001 Foreign order desk Telex 640055

ROM SUBROUTINES

Mark Zimmermann

In order to hinder their competitors' efforts to rip off their excellent BASIC interpreter, MICROSOFT (and Commodore) have not released any significant information about the BASIC used in the PET microcomputer. If one wants to write machine-language programs which do floating-point mathematics, one apparently has to write all of one's own routines. This is not only hard to do; it also wastes space in RAM, and is particularly frustrating when one just wants to call a simple function available in BASIC.

In this note, I'd like to share some addresses I've discovered in the PET ROMs which are useful for executing single-argument functions: SQR, RND, LOG, EXP, etc. I'd also like to make an appeal for information from readers who have figured out how to do two-argument (dyadic) functions such as +, *, /, and $\uparrow$.

As most PET machine-language users know by now, MICROSOFT BASIC stores floating-point numbers using 32 bits for the fractional part and an 8-bit exponent, with a trick to include the sign bit in the fractional part. (See PEOPLE'S COMPUTERS, Vol. 7, No. 2 (Sept.-Oct. 1978) pages 16-19 for a more detailed description.) When doing calculations, numbers are moved to the page zero "floating accumulator" in memories B0-B5, and the sign bit is unfolded from the fractional part. B0 holds the exponent of the floating-point number with 80 added to it (128 decimal added); the binary fraction, normalized to be between 1/2 and 1, is stored in B1, B2, B3, and B4, with radix point at the left; and bit 7 of B5 is the sign bit of the floating-point number.

If a user-developed machine-language program uses the same format, puts a number into B0-B5, and then calls a PET ROM subroutine, when control returns from the subroutine the result of the function called will be returned in the floating accumulator. That's all one needs to know to use the table of ROM subroutines which follows!

For example, to calculate $\exp(-u^2)$ via ROM subroutines (for nonzero u) the following sample assembly-language program will do the trick:

```
JSR DB2A    ;ABS(u)
JSR D8BF    ;LOG
CLC
LDA #1
ADC B0
STA B0      ;adding 1 to exponent doubles number
JSR DEA0    ;EXP (so now we have u*u)
LDA #FF
EOR B5
STA B5      ;flip sign bit
JSR DEA0    ;EXP(u*u)
RTS
```

(if any of the reduced listings can't be read, send a couple of stamps to Box 379 for full-size copies.)

If this program is called using the USR(U) function, the PET's BASIC will set up U in the floating accumulator for you, and will re-translate and store (or print, or whatever you want) the result for you after control leaves the USR function. Unfortunately, it turns out to take about twice as long to calculate $\exp(-u^2)$ this way as it does to just define a BASIC function: DEF FNG(X)=EXP(-X*X). The problem, of course, is that to square u , we used the LOG and EXP functions instead of simply multiplying $u*u$! So, if anyone can tell me where to find and how to use the simple floating-point operations like addition and multiplication, I'll be most grateful!!

I found the addresses given in Table 1 while prowling through the PET's ROMs. In particular, the code following address CCB8 which evaluates BASIC expressions led me to a table following address BFDE. As many of you probably know, PET BASIC stores key words (such as EXP, COS, ASC, etc.) as single-byte "tokens". (Table 1 includes the hexadecimal values of the tokens in column 4.) PET's BASIC doubles the value of the token (throwing away any carry) and puts that into its Y register. It then looks up the address stored in BFDF+Y and BFDE+Y. That's the address of the function subroutine to be executed!

I'd like to acknowledge the help of Jim Butterfield, who compiled token lists and gave me a wonderful disassembler.

Copyright © 1979 Mark Zimmermann

PET ROM SUBROUTINES - - TABLE 1

Function	Address	Decimal	Token	Decimal
SGN	DB0B	56075	B4	180
INT	DB9E	56222	B5	181
ABS	DB2A	56106	B6	182
USR	0000 (:) :	0	B7	183
FRE	D264	53860	B8	184
POS	D285	53893	B9	185
SQR	DE24	56868	BA	186
RND	DF45	57157	BB	187
LOG	D8BF	55487	BC	188
EXP	DEA0	56992	BD	189
COS	DF9E	57246	BE	190
SIN	DFA5	57253	BF	191
TAN	DFEE	57326	C0	192
ATN	E048	57416	C1	193
PEEK	D6E6	55014	C2	194
LEN	D654	54868	C3	195
STR\$	D549	54089	C4	196
VAL	D685	54917	C5	197
ASC	D663	54883	C6	198
CUR\$	D5C4	54724	C7	199
LEFT\$	D5D8	54744	C8	200
RIGHT\$	D604	54788	C9	201
MID\$	D60F	54799	CA	202

Two other useful subroutines, courtesy of SPHINK & PUG:

D0A7 (53415) converts floating-point to integer in B3, B4 (high, low)
D278 (53880) converts integer in A, Y to floating-point

continued.....

example		all numbers in hex below this line	
S	P	contents	S ← P ← s/2
B8	B0	exponent-80	82 81
B9	B1	fraction MSB	AD C9
BA	B2	frac. byte 2	F8 0F
BB	B3	frac. byte 3	54 DA
BC	B4	fraction LSB	59 A2
BD	B5	sign	00 FF
BE		sign comparison	FF
BF		roundoff byte	00

0000 USR(P)
 CED6 S OR P
 CED9 S AND P
 D06A FRE(P)
 D285 POS(P)
 D349 STR\$
 D5C4 CHR\$
 D5D8 LEFT\$
 D604 RIGHT\$
 D60F MID\$
 D65A LEN
 D663 ASC
 D685 VAL
 D6E6 FEEK
 D728 S ← P
 D73C JSR D95E, then S ← P
 D73F S ← P
 D7AC normalize P
 D8BF LOG(P)
 D8FD JSR D95E, then S ← P
 D900 S ← P
 D95E [A+100Y]←S, separating sign, set sign comparison, return with (B0) in A
 D9EA S / P
 DA7h [A+100Y]←P, separating sign
 DAAE roundoff(P)-[X+100Y], merging sign
 DACE S←P
 DAEF P←S, with rounding
 DAE1 P←S, without rounding
 DAE0 roundoff(P)
 DAFD SGN(P)←A
 DB0B SGN(P)
 DB2A ABS(P)
 DB9F INT(P)
 DE2h SQR(P)
 DE2E S ↑ P
 DEAO EXP(P)
 DF45 RND(P)
 DF9E COS(P)
 DFA5 SIN(P)
 DFEE TAN(P)
 E048 ATN(P)

Note: +, -, *, /, and ↑ must be called with S in B8-BD, sign comparison set to (B5) EOR (BD), and (B0) in A

Constants in floating (merged sign) notation:
 E024 1/4
 DDE3 1/2
 E01A π/2
 CDBC π
 E01F 2π

Thanks to Jim Butterfield, Gene Beals' PET USER GROUP, and SPHINX for helpful data.

conversions: D0A7 P←integer in B3,B4 [100B5+B4]
 D278 integer in A,Y [100A+Y] ← P

MACHINE LANGUAGE GUIDE

NOW AVAILABLE in a single source from ABACUS SOFTWARE is a comprehensive guide to the inner workings of your PET Computer. This guide will show you how to access many of the operating system and BASIC routines from a machine language program.

Learn how to use:

- * input and output routines from the keyboard and video display
- * clocks and timers for high precision interval measurement
- * floating point arithmetic and number conversion
- * builtin arithmetic functions
- * machine language subroutine linkage and hints from BASIC programs

Available for \$5.95 from ABACUS SOFTWARE. VISA/BA or MASTERCARD cards are accepted if you include your card number and expiration date. Postage paid in U.S.

ABACUS SOFTWARE
P.O. BOX 7211
GRAND RAPIDS, MI. 49510

QUALITY SOFTWARE FOR THE PET*

*PET is a product of Commodore Business Machines, Inc.

What does quality software mean? First, it means programs that work the way you'd like them to, easily and naturally. It means internal operating instructions and internal documentation, as well as system design and modular construction to ease modification or maintenance. Finally, quality software is provided on quality tapes (and with MICRO SOFTWARE SYSTEMS, it means a quality, hard plastic case for the cassette!).

NEW! Many added features... kids love it!
SWEET-PIG™ a simple but captivating bouncing-ball game. The number of ways it can be played is limited only by the imagination of the player. Same old price. \$5.95

METRIC-CALC™ turns the PET into a powerful stack-operated RPN calculator with scientific functions, metric conversions. 20 levels, 20 memories. Too much to describe. \$7.95
 Write for flyer!

NEW! Our most popular... it's SUPER!
CHEQUE-CHECK™ end the hassle in getting checkbook and bank statement to balance. Review or change entries, verify checkbook line-by-line. Even wives use it! \$7.95

MEM-EXPLORER™ can be combined automatically with existing programs, shows exactly how and where they are stored in memory. Described in The ____ Paper articles, by Roy Busdiecker. \$7.95

BUSINESSMEN NOTE: Compare this continuous display to the cost of a single ad!
BILLBOARD™ lets you type message up to 254 characters (incl. pause, flash), which is then displayed in 1" letters moving horizontally. Repeats continuously. No static, snow. Good sales display! \$49.95
 * You set the speed, too.

MICRO SOFTWARE SYSTEMS

P.O. Box 1442, Woodbridge, VA 22193

Send check or money order. Allow two weeks for check to clear. VA residents add 4% tax. Dealer inquiries invited.

MACHINE LANGUAGE ROUTINES FOR FAST GRAPHICS

Warren Swan

Now you BASIC programmers, don't go flipping past this article. The machine language routines in this article do not have to be understood to be used. They merely let you do some things with your PET's display screen that you cannot do as fast in BASIC.

There are five machine language routines presented in this article. They consist of routines to

- toggle the screen on/off bit to turn the screen off or on
- fill the screen with any requested character
- inverse every character on the screen so that unreversed characters are reversed and reversed characters are unreversed
- scroll the screen down one 40 character line, instead of up
- swap the screen memory with 1K of RAM that is rendered inaccessible to BASIC, including the current graphics mode swapped with the stored graphics mode in the swap area.

A BASIC loader program, which can be used to load *any* machine language routine *anywhere* in unused RAM, is also presented and used to load the package of graphics machine language routines.

Listings 1 through 5 are the assembly language programs for the five routines. All routines but the Screen Filler routine are called from BASIC via the SYS command. The Screen Filler (listing 1) is called via the USR command.

To use these routines, type in the relocating loader program in listing 6 and RUN it. It will ask you where the base location at which you wish to load these routines is. Most likely, you will want to load them in a protected area of memory such as the buffer area for the second cassette recorder if you don't use one. If you wish to use the second cassette buffer, enter 826 for the base. If you want to load them at the top of memory and keep them from being mutilated by BASIC, execute the following statements:

```
POKE135,PEEK(135)-1:CLR  
PRINT PEEK(135)*256+PEEK(134)
```

Then RUN the program and enter the number printed by the second line above for the base.

At any rate, the loader program will load these routines into memory and print the absolute location that is the starting address of each routine. Write these locations down for later use. Ignore the PAGE and BYTE columns for now.

To use the Screen off/on routine, simply use SYS NNN; where NNN is the location you wrote down for this routine from the loader. When this command is executed, if the screen was on, it will turn off; if it was off, it will turn on. This routine can be used to turn off the screen while forming a graphics picture that you don't want the user of your program to see until it is finished.

To use the Screen Filler routine, simply use A=USR(NNN); where A is some dummy variable you don't mind having cleared by the routine, and NNN is the character number of the character you want to fill the screen. The routine will instantly fill the PET's screen with the desired character. If you are not familiar with the "character number" to which I refer, these are the

numbers (0 to 255) which you can POKE into the PET's screen RAM to display them. To find out what all 256 character numbers are for, execute the following command line:

```
FOR L=0 TO 255:POKE L+32768,L:NEXT
```

This routine can be used to fill the screen with a given character as a background to a graphic display. Or it can be used to create dazzling displays that cause the screen to appear to be moving.

To use the Screen Inverse routine, simply use SYS NNN; where NNN is the location you wrote down for this routine from the loader. When this command is executed, the screen will be "inversed". That is, every unreversed character will be reversed; and every reversed character will be unreversed. This routine may be used to cause a graphics picture to flash or blink.

To use the Screen Scroll-down routine, simply use SYS NNN; where NNN is the location you wrote down for this routine from the loader. When this command is executed, the lines of the screen will scroll down, just as they would scroll up when printing off the last line of the screen. This routine can be used to move entire pictures on the screen down, and with help from the PRINT statement, up.

To use the Screen Swapper routine, simply use SYS NNN; where NNN is the location you wrote down for this routine from the loader. The first call to this routine is only used to set up pointers and clear the "save area" - the memory locations that will contain the contents of the screen that are swapped into it. After the initial call the user must execute a NEW or CLR command. Otherwise, BASIC will continue to store strings in the save area.

The Screen Swapper routine may be used to create "moving" pictures. This is done by printing a graphic picture on the screen, calling the routine to swap it into the store area, printing another graphic picture, and then repeatedly calling the Swapper to swap back and forth between pictures. The Screen off/on routine may be helpful here, so that the "drawing" of the pictures is not visible until after they are done.

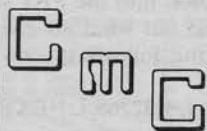
Now you're probably still wondering exactly how you can use all these routines to draw some really spiffy graphic pictures and animate them (unless you're super imaginative). So to help get you started, listing 7 is a demo program that uses all of the routines to show them off, even though I'm not particularly imaginative when it comes to graphics.

Need any more fast graphic routines? Drop me a line and let me know. I should have some free time this summer...

Warren Swan
15933 S. Grove Avenue
Oak Forest, IL 60452

also, just a quick explanation of the relocating loader program. All data is read in as string. Any string with a "/" at the right is taken as a routine name and is printed out, followed by the absolute location that the routine will be loaded at; followed by the page number and byte number for this location (e.g., location 826 (decimal) is at page 3, byte 58 because $826 = 3*256 + 58$).

A number string is converted to a number and loaded at the next location (starting from the BASE initially). If a number has a "*" attached to the right, it is added to the most recent routine starting address and the two byte address so formed is loaded into the next two locations,



CONNECTICUT microCOMPUTER, Inc.

150 POCONO ROAD - BROOKFIELD, CONNECTICUT 06804

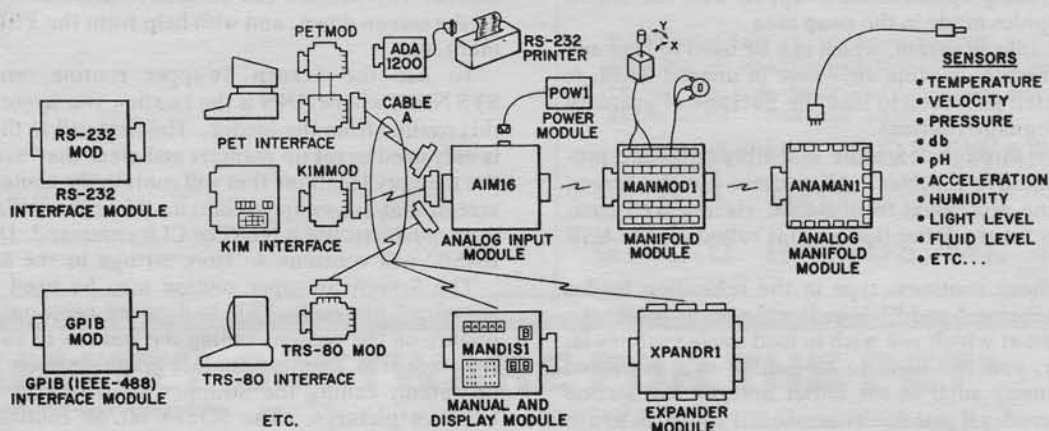
(203) 775-9659



New Product Announcement

PETMOD (PETS AVR)

Analog to Digital Converter Adapter Module
for the COMMODORE PET



The DAM SYSTEMS 16 channel Analog Input Module, the AIM16, can now be directly connected to the COMMODORE PET using the DAM SYSTEMS PETMOD. This system provides very easy to use and low cost analog input for the PET. Joysticks, potentiometers, temperature sensors, accelerometers, pH meters, etc. can now be connected to the PET with minimal effort.

The PETMOD plugs into the PET IEEE port and PET user port. The PETMOD then has available two PET IEEE ports and one user port in addition to the DAM SYSTEMS port.

The PETMOD is also called the PETS AVR. Once plugged in, it can be left in, and the PET owner can use the IEEE and user ports on the PETMOD saving the fingers on the printed circuit board of the PET from becoming discolored.

The PETMOD sells for \$49.50.

low byte followed by high byte. This is used to load absolute addresses given as offsets within the current routine.

If a number has a "!" attached to the right, it is added to the BASE address and the two byte address so formed is loaded into the next two locations. This is used to load absolute addresses given as offsets to the BASE location.

The signal for the end of data is the "#" character followed by a number which is added to the BASE address and the two byte address is stored in locations 1 and 2 for the USR function.

Copyright © 1979 by Warren Swan, Oak Forest, IL, USA

LISTING 1: Screen off/on routine.

```

AD11E8  *BASE
4908    SWITCH LDA $E811 ! Load 'PIAL1'.
8D11E8  BDR #0E ! Flip bit 3 (screen control).
60      STA $E811 ! Store 'PIAL1'.
        RTS

```

LISTING 2: Screen Filler routine.

```

20A7D0  *BASE+9
A000    FILLSCN JSR $DOA7 ! Get integer parameter.
8420    LDY #0 ! Initialize offset.
A280    STY $20 ! Screen pointer low byte.
A584    LDA $B4 ! Screen address high.
8624    LDY #80 ! Screen address high.
9120    LDY #80 ! Get parameter (Character #).
C8      NEXT STX $21 ! Screen pointer high byte.
DOFB    STA ($20),Y ! Store character in screen.
E8      INY ! Next location.
DOF8    BNE NEXT ! End of page? NO: Repeat.
E8      INX ! YES: Next page.
DOF4    CFX #84 ! Finished last page?
DOF4    BNE NEXT ! NO: Do next page.
60      WAIT RTS ! Now comes several options:
        or LDA $203 ! Test 'key pressed' location.
AD0302  BDR #FF ! If no key pressed:
        BEQ WAIT ! THEN wait for key to be pressed.
        RTS ! ELSE return.
        or change BEQ WAIT to BNE WAIT:
DOF9    BNE WAIT ! THEN return, ELSE wait for key to
        ! be released.

```

LISTING 3: Screen Inverse routine.

```

A280    *BASE+40
A000    INVRS LDX #80 ! Screen address high.
8422    LDY #0 ! Initialize offset.
8622    STY $21 ! Screen pointer low byte.
B121    LDY #80 ! Screen pointer high byte.
4980    NEXT LDA ($21),Y ! Get character from screen.
9121    BDR #80 ! Flip bit 7 (rvs/un-rvs bit).
C8      STA ($21),Y ! Store character back to screen.
DOF7    INY ! Next location.
E8      BNE NEXT ! End of page? NO: Repeat.
E8      INX ! YES: Next page.
DOF4    CFX #84 ! Finished last page?
DOF4    BNE NEXT ! NO: Do next page.
60      RTS ! YES: Return.

```

LISTING 4: Screen Scroll-down routine.

```

20      *BASE+63
A02E--  SCRDWN JSR SWITCH ! Turn off screen. Note SWITCH=BASE.
8422    LDY #38 ! Initialize offset. Destination is
A000    STY $22 ! 40 bytes beyond source.
8420    LDY #0 ! Set up source pointer
A0B8    LDY #0 ! low byte.
A283    LDY #BF ! Start @ last char. of 2nd to last
8621    LDY #BF ! line. Address high.
8623    LDY #21 ! Store pointers'
8623    STX $23 ! high bytes.
9120    NEXT LDA ($20),Y ! Get source byte.
9122    STA ($22),Y ! Put in destination location.
E8      DEY ! Next byte. End of page?
COFF    CFX #FF ! (i.e. - have we gone past zero?)
DOF7    BNE NEXT ! NO: Next byte.
CA      DEX ! YES: Next page.
DOF7    CFX #7F ! Last page done?
DOF7    BNE NEXT ! NO: Do next page.
A027    LDY #39 ! YES: Reset offset to 39 (decimal).
A920    LDA #20 ! Load accumulator with ASCII space.
9120    CLR STA ($20),Y ! Clear top line of screen.
E8      DEY ! Next byte. All 40 bytes done?
10FB    BFL CLR ! NO: Clear next location.
4C      JMP SWITCH ! Turn on screen and return.

```

Pointer to source location: Pointer to destination location:

\$20 \$21 \$22 \$23
 00 E3 (Initially) 28 83

(if any of the reduced listings can't be read, send a couple of stamps to Box 379 for full-size copies.)

LISTING 5: Screen Swap routine.

```

A200    *BASE+108
A993    SWAP LDX #00 !
FO05    LDA #147 ! Load flag value.
        BEQ SKIP ! If not first time through,
        ! skip initialization.
A587    LDA $87 ! Load high byte of pointer to top
38      SEC ! of memory. Subtract off 4 pages
E904    SBC #04 ! (1K bytes) of memory and
8587    STA $87 ! restore pointer. BASIC now has
        ! access to 1K less memory.
A90C    LDA #12 ! Set initial default
8186    STA ($86,X) ! mode to graphics.
20----- SKIP JSR SWITCH ! Turn off screen.
A186    LDA ($86,X) ! Swap current
AC4CE2  LDY $E84C ! graphics mode
8D4CE6  STA $E84C ! byte with previously
98      TYA ! stored graphics
8186    STA ($86,X) ! mode byte.
A486    LDY $86 ! Save area pointer low byte
C8      INY ! (should always be 1, unless bad
8420    STY $20 ! memory chips).
8622    STX $22 ! Screen area pointer low byte (zero).
A000    LDY #00 ! Initialize offset.
A587    LDY #87 ! Save area high byte (also pointer
8621    STX $21 ! to top of memory).
A280    LDY #80 ! Screen area pointer high byte ($80).
8621    STX $23 ! Set up for next page to screen.
8121    LDA ($20),Y ! Get stored byte.
48      PHA ! Save temporarily.
B122    LDA ($22),Y ! Get screen byte and
9120    STA ($20),Y ! store it in save area.
68      PLA ! Recall stored byte
9122    STA ($22),Y ! and store it in screen.
C8      INY ! Next byte. End of page?
DOF3    BNE NEXT ! NO: Do next byte.
3621    INC $21 ! YES: Increment pointers to
E8      INX ! next page. Finished last page?
DOF4    CFX #84 ! (last page of screen is $8000):
DOF4    BNE NEXT ! NO: Do next page.
A1----- LDA SWAP+3 ! YES: Get flag again. Print it.
20D2FF  JSR $FFDE ! (First time it's 'clear screen').
8C----- STY SWAP+3 ! Zero flag for subsequent calls.
4C----- JMP SWITCH ! Turn on screen and return.

```

The first time the user calls SWAP (via the SYS command) it initializes the Top-of-Memory pointer for BASIC and clears out the save area. It then swaps the current screen contents with the save area and sets the graphics mode to graphics (not lower case). After the first call to SWAP, the user must execute a NEX or CLR command so that BASIC will not put string data into the save area.

NOTE: This code may not be used in Read Only Memories. It is "mixure code" which modifies itself.

LISTING 6: Relocating Loader for Machine Language Routines in BASIC.

```

0 REM: PACKAGE LOADER VERSION 1.1 - GRAPHICS 1.4 PAC BY WARREN D. SWAN
1978-1979
10 INPUT "BASE=";BASE
20 DATASWITCH/,173,17,232,73,8,141,17,232,96
30 DATASCREEN FILLER/,32,167,208,160,0,132,32,162,128,165,180,134,
  ,33,145,32
40 DATA200,208,251,232,224,132,208,244,96,3,2,73,255,240,249,96
50 DATASCREEN INVERTER/,162,128,160,0,132,33,134,34,177,33,73,128,
  ,145,33,200
60 DATA208,247,232,224,132,208,240,96
70 DATASCROLL DOWN/,32,0,160,40,132,34,160,0,132,32,160,191,162,131,
  ,134,33,134
80 DATA35,177,32,145,34,136,192,255,208,247,202,224,127,208,236,160,
  ,39,169,32
90 DATA145,32,136,16,251,76,0!
100 DATASCREEN SWAP/,162,0,169,147,240,11,165,135,56,233,4,133,135,
  ,169,12,129
110 DATA134,32,0,161,134,172,76,232,141,76,232,152,129,134,164,134,
  ,200,132
120 DATA32,134,34,160,0,166,135,134,33,162,128,134,35,177,32,72,177,
  ,34,145
130 DATA32,104,145,34,200,208,243,230,33,232,224,132,208,234,173,3*,
  ,32,210
140 DATA255,140,3*,76,0!
990 DATA NEXT AVAILABLE LOCATION/,#,9
1000 B=BA:I=0:PRINT "BASE=BA TAB(30)"PAGE BYTES
1010 READ N$:IF N$="# THEN READ N=BA+N:GOSUB 2000:POKE I,L:POKE 2,H:
  GOTO 1080
1020 N=VAL(N$):R$=RIGHT$(N$,1):IF R$="/" THEN B=B+I:I=0:N=B:GOSUB 2000:
  GOTO 1060
1030 IF R$="*" THEN N=N+B:GOSUB 2000:POKE B+1,L:POKE B+1,H:I=I+2:GOTO 1010
1040 IF R$="!" THEN N=N+B:GOSUB 2000:POKE B+1,L:POKE B+1,H:I=I+2:GOTO 1010
1050 POKE B+1,N:I=I+1:GOTO 1010
1060 N=LEN(N$)-1:IF N=23 THEN N=23
1070 PRINT LEFT$(N$,N$)":"B"TAB(30)"H"TAB(34)L:PRINT:GOTO 1010
1080 PRINT "USR:"N,B-B"BYTES TOTAL":END
2000 H=INT(N/256):L=N-H*256:RETURN

```

ss is clear screen, sd is cursor down, sl is cursor left

(listing 7 on p. 19)

SINGLE—STEP TRACE ROUTINE

Jim Russo

MT6671 is a machine-language monitor program incorporating a single-step trace feature. It occupies locations 1A00-1FFF, and is started by SYS 6671. It includes all the commands of the Commodore Monitor, plus a Disassemble command and a Trace command. The R command is used first to set-up the registers and program counter. The T command executes just one instruction of the user program, then displays the new contents of the registers (SR,AC,XR,YR,SP,PC) followed by the instruction at the PC and its disassembly. At this point the user may depress the RVS key to continue at about 2 instructions per second, the SPACE key to continue as fast as the PET can print the registers, the less-than key to step through one instruction at a time, or the STOP key to return to the monitor. A sample trace is included as Figure 1.

Notice that the instructions are displayed before they are executed. If you wish to change the flow of a program, you can STOP, use the R command to modify registers, or the M command to change memory, and then the T command to resume tracing.

There are some limitations on the use of the trace. Page zero locations 0A-22 and 40-47 are used, so beware of conflicts with the program you're tracing. Interrupts are used to step the program. If the user program changes the interrupt vector (219-21A) or the interrupt enables, the PET may crash.

The major portion of MT6671 is the Commodore Monitor, relocated to 1A0F-1D6A. It also includes the 6502 disassembler program, which has been around longer than the PET, at locations 1E00-1FE3. The part which I contributed is the trace feature, which includes little bits of code tucked in wherever there was room.

The execution of a single instruction is accomplished by setting the interval timer in the VIA to interrupt during the instruction. When the interrupt occurs, the PET saves the register contents, and jumps through the interrupt vector (219-21A) back to MT6671.

To illustrate how this is done, I have patched a simple single-step routine onto the Commodore Monitor. A complete listing of the Monitor is in the PET MANUAL. The patch is shown in Figure 2.

The routine which recognizes commands is modified to include T (498 and 76B-76E). When the T command is recognized, the interrupt vector is changed (776-77D) to point to 795. The interval-timer interrupt is enabled and the 60Hz interrupt disabled (780-785). The timer is set (78A-78F) for the period of time necessary to execute the remaining instructions in the monitor up to the RTI which effects the jump to the user program. The the GO routine in the monitor is used to load the registers and jump to the user program. When the interrupt occurs, control returns to the subroutine call at 795. A ROM subroutine is used to restore the interrupt vector and interrupt enables for the normal 60Hz interrupt service. Interrupt is turned back on (798), a bit set so the monitor won't decrement the PC (799), and control is returned to the monitor BRK routine which displays the registers.

This illustration program doesn't have all the features of MT6671, but it does illustrate how the task of controlling program execution one instruction at a time is accomplished.

MT6671 is available through the PET User Group. Future revisions will incorporate additional features. It will also be available relocated to other areas of memory, to allow its use with other programs that reside in 1A00-1FFF. Correspondence from interested users is welcome.

Jim Russo, 1421 Pine Valley, Ann Arbor, MI 48104

Figure 1
Sample of trace output

SR	AC	XR	YR	SP	PC	OP-CODE	DISASSEMBLY
A0	P1	81	00	F9	1E15	A1 44	LDA (\$44,X)
A0	85	81	00	F9	1E17	A8	TAX
A0	85	81	85	F9	1E18	4A	LSR
21	42	81	85	F9	1E19	90 08	BCC \$1E26
21	42	81	85	F9	1E18	4A	LSR
20	21	81	85	F9	1E1C	B0 17	BCS \$1E35
20	21	81	85	F9	1E1E	C9 22	CMR = \$22
A0	21	81	85	F9	1E20	F0 13	BFQ \$1E35
A0	21	81	85	F9	1E22	29 07	AND = \$07
20	01	81	85	F9	1E24	09 80	ORA = \$80
A0	81	81	85	F9	1E26	4A	LSR
21	40	81	85	F9	1E27	AA	TAX
21	40	40	85	F9	1E28	0E FE 1E	LDA \$1EFE,X
21	62	40	85	F9	1E2B	80 04	BCS \$1E31
21	62	40	85	F9	1E31	29 0F	AND = \$0F
21	02	40	85	F9	1E33	00 04	BNE \$1E39
21	02	40	85	F9	1E39	AA	TAX
21	02	02	85	F9	1E3A	86 42 1F	LDA \$1F42,X
A1	81	02	85	F9	1E3D	85 40	STA \$40
A1	81	02	85	F9	1E3F	29 03	AND = \$03
21	01	02	85	F9	1E41	85 41	STA \$41
21	01	02	85	F9	1E43	96	TYA
A1	85	02	85	F9	1E44	29 8F	AND = \$8F
A1	85	02	85	F9	1E46	AA	TAX
A1	85	85	85	F9	1E47	96	TYA
A1	85	85	85	F9	1E48	A0 07	LDY = \$03
21	85	85	03	F9	1E4A	00 8A	CPX = \$8A
A0	85	85	03	F9	1E4C	FC 0E	BFQ \$1E59

Figure 2
Patches to Commodore MONITOR to allow single-step execution.

```

076E CA      DEX
076C 30 03   BMI $0771      } replaces instructions
076E 4C 84 04 JMP $0484      } moved from 0498
0771 C9 54   CMP = $54      } test for T command
0777 C0 B9   BNE $072E
0775 78      SEI            } Interrupt off
0776 A9 07   LDA = $07
0778 8D 1A 02 STA $021A      } set interrupt vector
077E A9 95   LDA = $95      } to 0795
077C 8D 19 02 STA $0219
0780 A9 A0   LDA = $A0
0782 8D 4E E8 STA $E84E      } Interval timer enable
0785 CF 13 E8 DEC $E613      } 60Hz interrupt disabled
0788 A9 27   LDA = $27
078A 8D 48 E8 STA $E848      } Timer set for 39 microseconds
078D A9 00   LDA = $00
078F 8D 49 E8 STA $E849      } Timer starts counting here.
0792 4C E8 05 JMP $05E8      } Load registers & go
0795 2D FB FC JSR $FCFB      } Set interrupt to normal
0798 58      CLI
0799 4A      LSR
079A 4C 2E 04 JMP $042B      } Display registers.

.
.
READY.

.
.
.

0498 4C 6B 07 JMP $076B      } Jump into patch

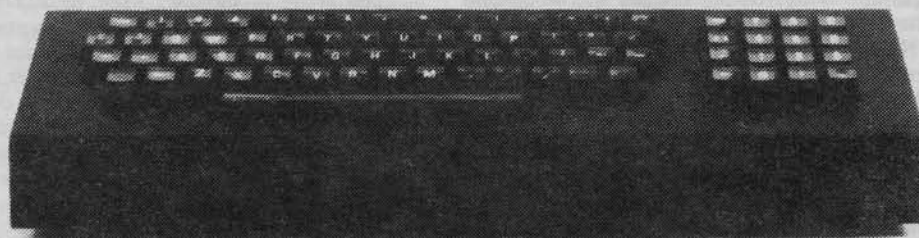
.
.
.

041D A9 9D   LDA = $9D      } New end-of-program value.

.
.
.

```

You love your PET, but wouldn't you love it more with this BigKeyboard?



The Skyles Big KeyBoard™. More than 15 inches wide. A layout nearly identical to the PET Keyboard and with *all* functions—alpha, numeric, graphics, special symbols, lower case alpha—on full-sized, almost plump, key-tops double-shot to guarantee lifetime durability.

Actual size

S

The switching action uses gold cross point contacts; the torsion springs are gold-plated. The housing is black anodized heavy aluminum. The keyboard assembly and housing have drain-holes; to clean the keys of "gook," rinse under clear water and let dry.

Parallel entry and unbelievably easy to install. Connect the PET's keyboard cable to the Skyles Big KeyBoard. Connect the Skyles cable connector to the main board and to the Skyles Big KeyBoard housing. Now you can use the Skyles for fast, finger-easy data entry, the original PET Keyboard for hunt-and-peck graphics entry. And you can disconnect either keyboard—or both—without reopening your PET: just remove the connector(s) from the Skyles housing.

Fully guaranteed, of course: • Use it for ten days. If you don't like it for any reason, return it for full refund; no questions asked • Guaranteed for 120 days; we'll repair or replace at no charge if the Big KeyBoard proves defective for any reason short of malicious mischief.

Order the Skyles Big KeyBoard now. Enjoy it within days.

SPRING OFFER

Be the first on your block to own a 16K PET with a Big KeyBoard.

And save money while you're doing it!

8KB Memory

Expansion System	\$250
74KB Big KeyBoard	125
shipping & handling	8.50
regular cost	\$383.50

SPRING OFFER \$350 COMPLETE!

(Or you can reach to 24k:)

16KB Memory

Expansion System	\$450
74KB Big KeyBoard	125
shipping & handling	10
regular cost	\$585

SPRING OFFER \$525 COMPLETE!

**California residents add 6-6½% sales tax where applicable*
VISA, MASTERCHARGE ORDERS CALL (800) 227-8398 (except California residents)

SPRING OFFER EXPIRES JUNE 21, 1979



Skyles Electric Works

10301 Stonydale Drive
Cupertino, CA 95014
(408) 735-7891

VISA, MASTERCHARGE ORDERS CALL (800) 227-8378 (except California residents)

PLOT

Jerry A. Velders

Here is a program which I call PLOT, which allows high resolution (50 x 80) plotting, driven by machine language subroutine, to be easily incorporated into BASIC programs. This routine resides in the second cassette buffer, where it is safe from commands such as LOAD and NEW, and it plots somewhat more quickly than an equivalent BASIC routine. The idea and basic logic of the program came from an article by F. J. Campbell which appeared in the User Notes in Vol. 1, Issue 3. Although PLOT is pretty fast, it is apparent that most of the time used in iterative plotting programs (like a sine wave) is spent doing mathematical operations in BASIC. In any case, PLOT is about three times faster than the BASIC equivalent, and was fun to write. Maybe someone out there can benefit from it.

I've included an assembly object listing, tediously typed by hand because I don't have a printer yet, and a BASIC routine which pokes the code into memory. The program works because Commodore provided us with a set of 16 graphics characters which allow division of each pixel into four quadrants, each of which may be manipulated independently of the others. By assigning one bit of a four bit binary number to each quadrant, and using 1 and 0 to represent on and off, the logical OR function can be used to make decisions about which character to put into each pixel.

The basic task is therefore to accept values for X (horizontal displacement) and Y (vertical displacement), determine which pixel they refer to, determine which quadrant of the pixel they refer to, and OR this value with the value of the current contents to find the location of the next value in the table.

For ease of understanding, the program can be dissected into several discreet sections, as follows:

033A-0346

initialization; loads data from TABLE into zero page (put there to optimize speed)

0347-0356

TABLE; these values are data, representing the various graphic symbols associated with high density plotting

0357-0390

picks up values of X and Y from zero page (where put by BASIC) and determines in which quadrant of pixel the new square belongs

0392-03CC

finds the screen location under consideration, as determined by X and Y, and puts the address in locations 1 (low) and 2 (high)

03CE-03E6

finds the right character from the TABLE and puts it into the right location...then returns to BASIC

The best way to understand the operation of the program is to read closely through the comments in the listing, checking out the paths it will take under various

values of X and Y. Please note that there are no built-in safeguards for faulty values of X and Y. X must be from 0 to 79, and Y from 0 to 49. A mistake here probably won't make the PET crash, but can make it try to poke its way through memory, and it makes sense to be careful.

The areas in memory used by PLOT consist of the second cassette buffer (for program storage), the tail end of the input buffer in zero page (used for the table and for X and Y), and locations 1 and 2, used for the low and high address of the screen location being dealt with. You shouldn't need to re-initialize the routine (SYS826) unless there's a significant use of the input buffer, like an input of more than 50 characters or so, or an input from an external device like the cassette.

```
10 REM LOADS MACHINE LANG. PLOTTING
11 REM ROUTINE INTO SECOND CASSETTE BUFFER
12 REM TO USE, INITIALIZE WITH 'SYS826'
13 REM
14 REM POKE68,(X VALUE ..0-79)
15 REM POKE69,(Y VALUE ..0-49)
16 REM
17 REM THEN 'SYS855'
18 REM
19 REM FOR EXAMPLE:
20 REM
21 REM 10 SYS826
22 REM 20 FORX=0T049
23 REM 30 Y=X
24 REM 40 POKE68,X:POKE69,Y
25 REM 50 SYS855
26 REM 60 NEXTX
27 REM
28 REM
80 FORZ=826T0998:READA$
90 A$=RIGHT$("00"+A$,2):B=0
95 FORX=1T02:C$=MID$(A$,X,1)
100 IFC$="A"ANDC$<="F"THENC=ASC(C$)-ASC("A")+10
110 IFC$="0"ANDC$<="9"THENC=ASC(C$)-ASC("0")
120 IFX=1THENB=B+C*16:GOTO140
130 B=B+C:POKEZ,B
140 NEXTX
150 NEXTZ
160 END
200 DATA A2,00,BD,47,03,95,4A,E8,E0,10
210 DATA D0,F6,60,20,6C,7C,E1,7E,7F,E2
220 DATA FB,7B,62,FF,FE,61,FC,EC,A0,18
230 DATA A5,44,09,FE,69,01,85,46,18,A5
240 DATA 45,09,FE,69,01,85,47,18,A5,46
250 DATA C5,47,D0,0B,A5,46,D0,13,A9,02
260 DATA 85,48,4C,92,03,18,A5,46,D0,0E
270 DATA A9,01,85,48,4C,92,03,A9,08,85
280 DATA 48,4C,92,03,A9,04,85,48,46,44
290 DATA 46,45,38,A9,18,E5,45,85,47,A5
300 DATA 44,85,46,A9,00,85,01,A9,80,85
310 DATA 02,A0,00,A5,47,F0,12,18,A5,01
320 DATA 69,28,85,01,A5,02,69,00,85,02
330 DATA C8,C4,47,D0,EE,18,A5,01,65,46
340 DATA 85,01,A5,02,69,00,85,02,A2,00
350 DATA A1,01,D5,4A,F0,06,E8,E0,10,D0
360 DATA F7,60,8A,05,48,AA,B5,4A,A2,00
370 DATA 81,01,60
READY.
```

There are undoubtedly ways to increase the speed of execution of this program. I'm a relative newcomer to computing and don't always know the optimum techniques to use. This is especially true in the screen address calculations in this program, in which I used successive additions of 40 to determine the screen line. I would be appreciative of any suggestions which would enhance either the speed or elegance of the program.

continued.....

By the way, I'm composing this article on the Word Processor sold by Connecticut Microcomputer, which is a very versatile tool for handling text. Although I don't have a printer, a friend has a Diablo which I used to test the RS232 interface that I bought from CMC to accompany the Word Processor. When the PET and the Diablo were connected, everything worked perfectly, and it was a little ironic to see an \$800 computer driving a \$3300 printer. Although we were just chugging along at 300 baud, it was really neat to have the PET write out some listings for me.

```

      FLOT
033A  A2 00      LDY #500      ! clear X reg: used as counter
033C  B7 47 03   LDA $0347,X  ! get first byte from TABLE
033F  95 4A      STA $4A,X    ! store in zero page (input buffer)
0341  18        INX          ! increment counter
0342  E0 10      CPX #510     ! done yet? (16 values to move)
0344  D0 F6      BNE $033C    ! if not done, go get another
0346  60        RTS          ! if done, return to BASIC
0347  20 6C 7C E1 ! TABLE
034B  7E 7F E2 FB ! TABLE
034F  7B 62 FF FE ! TABLE
0353  61 FC EC A0 ! TABLE

0357  18        CLC          ! get ready to add
0358  A5 44      LDA $44      ! get value of X left by BASIC &
035A  09 FE      ORA #3FE     ! test bit zero to determine if
035C  69 01      ADC #301     ! even or odd.
035E  85 46      STA $46      ! store result (0 if odd; 255 if even)
0360  18        CLC          ! get ready to add
0361  A5 45      LDA $45      ! get Y value and do the same thing
0363  09 FE      ORA #3FE     !
0365  69 01      ADC #301     !
0367  85 47      STA $47      !
0369  18        CLC          !
036A  A5 46      LDA $46      ! get previously stored values and
036C  C5 47      CWF $47      ! compare to see if both are even or
036E  D0 05      BNE $037B    ! both odd...if not goto 037B
0370  A5 46      LDA $46      ! if both same, are they odd?
0372  D0 13      BNE $0387    ! if both even goto 0387
0374  A9 02      LDA #502     ! if both odd, then it's quadrant 2
0376  85 48      STA $48      ! store quadrant number
0378  4C 92 03   JMP $0392    ! jump to next section
037B  18        CLC          ! just to be safe
037C  A5 46      LDA $46      ! get value
037E  D0 05      BNE $038E    ! if X even and Y odd then branch
0380  A9 01      LDA #301     ! otherwise, Y even & X odd. Store
0382  85 48      STA $48      ! 1 as quadrant number
0384  4C 92 03   JMP $0392    ! jump to next section
0387  A9 08      LDA #8       ! X and Y both even. Store quadrant
0389  85 48      STA $48      ! number ( 8 )
038B  4C 92 03   JMP $0392    ! jump to next section
038E  A9 04      LDA #304     ! quadrant is number 4
0390  85 48      STA $48      ! store quadrant number
0392  46 44      LSR $44      ! divide X by 2
0394  46 45      LSR $45      ! divide Y by 2
0396  38        SEC          ! get ready to subtract
0397  A9 18      LDA #24      ! load acc with 24(dec) # of scrn lines
0399  E5 45      SBC $45      ! subtract vert. displacement (Y/2)
039B  85 47      STA $47      ! store
039D  A5 44      LDA $44      ! get horiz. displacement (X/2)
039F  85 46      STA $46      ! store
03A1  A9 00      LDA #300     ! put zero in acc; 1st scrn addr. low
03A3  85 01      STA $01      ! store in loc. 1
03A5  A9 80      LDA #80     ! load 1st scrn addr. high
03A7  85 02      STA $02      ! store in loc. 2
03A9  A0 00      LDY #500     ! clear Y reg
03AB  A5 47      LDA $47      ! load acc. with line number
03AD  F0 12      BEQ $03C1    ! if zero, goto next part
03AF  18        CLC          ! ready to add
03B0  A5 01      LDA $01      ! get low address of scrn loc
03B2  69 28      ADC #328     ! add 40 (dec)
03B4  85 01      STA $01      ! put it back
03B6  A5 02      LDA $02      ! get scrn address high
03B8  69 00      ADC #500     ! add 0 (insures carry)
03BA  85 02      STA $02      ! put it back
03BC  08        INY          ! increment Y reg
03BD  C4 47      CFI $47     ! are we at the right line yet?
03BF  D0 EE      BNE $03AF    ! if not go back and do it again
03C1  18        CLC          ! ready to add
03C2  A5 01      LDA $01      ! get low scrn addr
03C4  65 46      ADC $46      ! add appropriate X value
03C6  85 01      STA $01      ! put it back
03C8  A5 02      LDA $02      ! get high scrn addr
03CA  69 00      ADC #500     ! add zero to insure carry
03CC  85 02      STA $02      ! put it back (locs.1&2 point to scrn
03CE  A2 00      LDX #300     ! loc where character goes) clear X reg
03D0  A1 01      LDA ($01,X)  ! get contents of scrn location
03D2  D5 4A      CMP $4A,X    ! compare with first val in TABLE
03D4  F0 06      BEQ $03DC    ! if equal then goto 03DC
03D6  E8        INX          ! if not equal increment counter
03D8  E0 10      CPX #510     ! checked all 16 yet?
03DA  D0 F7      BNE $03D2    ! if all checked, but no match is made,
03DC  60        RTS          ! then return to BASIC..something wrong.
03DD  8A        TXA          ! get position of matched val in TABLE
03DE  05 48      ORA $48      ! logical OR with quadrant number
03DF  AA        TAX          ! put result in X
03E0  B5 4A      LDA $4A,X    ! get corresponding value from TABLE
03E2  A2 00      LDX #300     ! clear X
03E4  81 01      STA ($01,X)  ! put value into screen location
03E6  60        RTS          ! and return to BASIC

```

FAST GRAPHICS, ctd. from p. 15.....

LISTING 7: Graphics Pac Demo Program

```

0 REM: GRAPHICS PAC DEMO PROGRAM BY WARREN D. SWAN
10 BASE=826:SI=BASE+40:SD=BASE+63:SW=BASE+108:GR=59468
20 POKE GR,12:AS="FAST GRAPHIX BY MACHINE LANG. ROUTINES
30 PRINT"&":FORI=1TO25:SYS SI:SYS SD:PRINT"&":AS:NEXT:GOSUB1000
40 GOSUB200:AS="Fast Graphix by Machine Lang<del>vs</del> off Routines
50 SYS SA:POKE GR,14:GOSUB200:FORI=1TO RND(7)*10+20:SYS SA
:FORJ=1TO200:NEXT
60 NEXT:FORI=0TO255:A=USR(1):NEXT:A=USR(RND(7)*256):FORI=1TO24
:SYS SD
70 GOSUB500:NEXT:PRINTB$:FORI=1TO24:PRINT:GOSUB500:NEXT
80 RESTORE:POKE GR,12:FORI=1TO4:FORJ=0TO7:READB(J):NEXT
90 FORK=0TO3:WAIT 59456,32,32:A=USR(B(K AND 7)):FORM=1TO20:NEXT
:NEXT:NEXT
100 GOTO10
200 TS="[ch,12cd]":B$="[ch,24cd]":PRINT"&":SYS SI
210 FORI=1TO13:SYS SI:PRINTLEFT$(TS,1)ASLEFT$(B$,26-1)AS:;NEXT
:GOTO1000
500 FORJ=1TO100:NEXT:RETURN
1000 FORI=1TO323:NEXT:RETURN
9000 DATA 101,84,71,66,93,72,89,103
9010 DATA 100,82,70,64,67,68,69,99
9020 DATA 103,69,72,93,66,71,84,101
9030 DATA 99,69,66,67,64,70,82,100

```

In this listing, ce means clear screen, ch means cursor home, rvs means the reverse key, off means shift of the reverse key, cd means cursor down. [ch,12cd] means a cursor home followed by 12 cursor down characters. [ch,24cd] means a cursor home followed by 24 cursor down characters.

PET

second cassette

FEATURES

- SANYO CASSETTE RECORDER WITH COUNTER
- CUEING—AUDIO LOCATION OF PROGRAMS
- PLAY AND RECORD SENSE
- INTERFACE MODULE PLUGS DIRECTLY INTO CASSETTE PORT

\$69.95 & \$3.00 P & H

INTERFACE MODULE SEPARATELY

\$24.95 & \$2.00 P & H

D & R CREATIVE SYSTEMS
P.O. BOX 402N
ST. CLAIR SHORES, MI 48080
DEALER INQUIRIES INVITED

RAMBLIN'

HOW THE CB2 LINE WORKS...

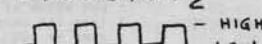
CB2 IS ONE CONNECTION (OF MANY), TO A SPECIAL CHIP IN THE PET CALLED THE VERSATILE INTERFACE ADAPTER (VIA), COMMODORE PART NUMBER 6522.

THE VIA APPEARS TO THE 6502 MICROPROCESSOR TO BE NOTHING MORE THAN A GROUP OF MEMORY ADDRESSES, NO DIFFERENT FROM ANY OTHER RAM REGISTERS; ACCEPTING AND RETURNING 8-BIT BINARY NUMBERS UNDER PROGRAM CONTROL. HOWEVER, INTERNAL CONTROL CIRCUITS IN THE VIA PERMIT A NUMBER OF NEAT THINGS TO HAPPEN.

FOR INSTANCE, VIA ADDRESS 59466 IS A SERIAL I/O SHIFT REGISTER.

IF YOU PUT DECIMAL 85 IN IT...
(POKE 59466, 85)

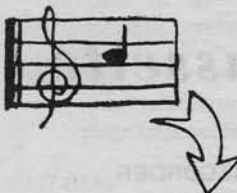
... IT WILL THEN CONTAIN
BINARY 01010101.

$85_{10} = 01010101_2$
CB2 SEES 

$51_{10} = 00110011$ (HALF THE FREQUENCY OF 85)

$15_{10} = 00001111$ (HALF AGAIN)

(YOU CAN SEE WHY 170_{10} (10101010_2)
SOUNDS THE SAME AS 85 (01010101)



	FREQUENCY Hz	SHIFT RATE 59464	BYTE SHIFTED 59466
-A	880	140	51
G	784	157	51
F	698	177	51
E	656	188	51
D	587	211	51
C	523	237	51
B	494	251	51
A	440	140	15
G	392	157	15
F	349	177	15
E	330	188	15
D	294	211	15
-C (FORMULA)	262	237	15
	$SR = \frac{62500}{F} - 2$	$\hat{F}$ $\downarrow$	$\hat{SR}$ $\downarrow$
		$\hat{15}$ $\downarrow$	

NOW, IF YOU POKE VIA ADDRESS 59467 WITH 16, IT WILL SET UP A FREE-RUNNING CONDITION, IN WHICH THE BITS IN 59466 ARE SHIFTED OUT "ENDWISE" ONTO THE CB2 LINE WHICH, IN OUR EXAMPLE, WILL CAUSE CB2 TO GO ALTERNATELY HIGH AND LOW AS THE ONES AND ZEROS GO BY. YOU CAN HEAR THIS AS A TONE THROUGH AN AMPLIFIER HOOKED UP TO CB2.

WHAT PITCH? WELL, THAT DEPENDS ON WHAT NUMBER IS POKED INTO 59464, A VIA REGISTER WHICH KEEPS TRACK OF THE TIME. THE BIGGER THE NUMBER, THE LOWER THE PITCH.

-O'B

MISCELLANY

Roy O'Brien

Well, everything else in this issue is put to bed; this column is the last thing. It's been an experience, and I can't imagine how Gene did it. I offered to help him, since I have a connection with a typing-typesetting-office services business which is pretty well equipped, but without their facilities it's gotta be a difficult job.

This time around, and probably next, the mailing labels are being done on a copier, but when we get one of the new printers, they'll be done via PET cassette files. Eventually, I hope to have a disc system up, to do both the mailing list and word processing. We have the CMC word processing program (which we've never had a chance to use) and will be looking for a mailing list program. Anybody got any recommendations?

We'll also be glad to have the printer to do the program listings, complete with cursor controls and graphics characters. Even in the big magazines, when a program is typeset, there are almost always errors, and you've got to be pretty experienced (as I am not) to debug 'em. It's hard to beat the idea of having a program up and running, checked out, then dumping a LIST to the printer. Meanwhile, as you can see, we're going with whatever the article author sent, as being more likely to be correct than a retyped version.

Incidentally, as it says elsewhere, if any of the listings are too small or unclear, drop a line to get full-size copies. One stamp per two-sided sheet will keep the books balanced.

A lot of the User Group mail contains comments or requests regarding machine-language information. We had planned the first of a tutorial series on machine code for this issue, but it got bumped (ironically) by a couple of really good machine-language programs. Look for it next time. I tend to share Gene's feelings on not including a lot of reviews, but will try to do one on the Abacus Machine Language Guide. If you've got it, I'd appreciate a postcard or note with your comments.

I recently got a chance to fulfill a long-time ambition, and attended the April SPHINX meeting. They meet in the Mr. Calculator store in downtown San Francisco, and it turned out to be the same sort of friendly pandemonium I've become accustomed to at the New Jersey Computer Club. Since many of the best programs in the NJ Club Library (and the Cassette Exchange) came out of SPHINX, I was glad to meet Jim Page, their tape librarian. We've agreed to get together soon for a full session of cross-pollinating our respective tape libraries.

While there, I picked up a copy of a flyer by Harry Saal (author of MAXIT, among other things), announcing an interesting product he's come up with. It's a 2K add-on ROM, which plugs into the expansion port, and provides the following additional commands to PET's BASIC ROM:

AUTO: enter autonumbering mode, with the PET providing evenly spaced line number prompts

APPEND: appends a BASIC program from tape to the program in storage—using normal SAVED tapes

DELETE: deletes a range of lines as easily as LIST

DUMP: displays the names and values of variables in the symbol table during or after running a program

HELP: used after an error in BASIC to display the erroneous line, with the offending taken highlighted

RENUMBER: rennumbers a BASIC program (and all references) by specified stepsize

TRACE: lets you see the line numbers of statements as they are executed, in a small scroll window in the corner of the screen. Also operates in single STEP mode

Harry calls it the BASIC PROGRAMMERS TOOL KIT, and is selling it for \$75. If you want more info, write Palo Alto IC's, 810 Garland Drive, Palo Alto, CA 94303.

Incidentally, the reason you haven't gotten a recent issue of the SPHINX Newsletter is that they've run into a temporary snag of some sort. Something to do with the funds being deposited with the University Regents and red-taped to a standstill. Milt Lee says they expect to get something going soon.

The Cassette Exchange (p. 3) contains KEYBOARD, a program originally written by Neil Harris to turn the PET keyboard into an organ, later modified by Jim Butterfield to include a record/playback feature. (By the way, we have a little instruction sheet with directions and cutout graphics to make an overlay for the PET keyboard which gives you the black and white keys in their proper positions to go with the program.) Anyway, the program aroused my ignorance of music to the point where I started shopping around for an instrument to fool with, which led to a lot of reading and discussion regarding synthesizers.

Turns out most of them have a simple tone generator, which produces the same kind of signal KEYBOARD puts out onto the CB2 line, which is then shaped (or bent, or broken, if you prefer) by analog circuitry. The shaping and filtering of the basic tone is triggered by a pulse emitted by the tone generator at the start of the tone. Well, if the PET can be made to put out a similar pulse on one of its other I/O lines at the start of a tone (as I'm sure it can), then a very simple rig will put you into the far-out music business. I've sent to PAIA for further info on their GNOME microsynthesizer, which sells for fifty bucks or so, and it looks to me as though it ought to hook up to the PET just fine. Will keep you posted.

The other side of the coin is that the whole analog end may be unnecessary, if you have a digital to analog (D/A) converter and appropriate software. Had a talk with Hal Chamberlin recently, much of which I couldn't understand, but Hal and others who are working in this area are looking forward to some interesting developments.

Speaking of developments, at the April NJ club meeting Joanathan Greif gave me a "preliminar" copy of a music program he's working up. Like KEYBOARD, you can record a tune on the PET keyboard, but when it plays back, the screen first shows a blank musical staff, complete with G-clef, and as each note is played, it pops into its proper place on the staff. Jon says he's going to call it MUSICREADER, but I think MUSICWRITER would be more appropriate. Boy, I sure hope Commodore starts shipping printers soon!

B

MEMORY TEST

Jerry A. Velders

MTEST is a non-destructive machine language memory testing routine which was built around the PET but which could be easily adapted to any 6502 based machine. In this version the test routine resides in the second cassette buffer (HEX 033A or DEC 826) and occupies 52 bytes of memory. I am enclosing a BASIC program which loads MTEST and then supports its operation, allowing user specification of starting address; and is easily modified to work on any amount of contiguous memory.

The BASIC program loads MTEST and then prompts the user for the starting location in decimal, after which it jumps via SYS 826 to the test routine. Each consecutive byte is loaded with 0 through 255 and checked to see that it reads back correctly. Prior to testing each byte, the original contents are saved in location 70(DEC) and after the test the contents are restored. This is what makes it possible to jump back to the BASIC program after the test is complete. Memory locations 1 and 2 (normally used by the USR statement) are used as indirect pointers to the byte being tested, so if a bad byte is encountered the test branches back to BASIC, saving the address of the bad location in 1 and 2.

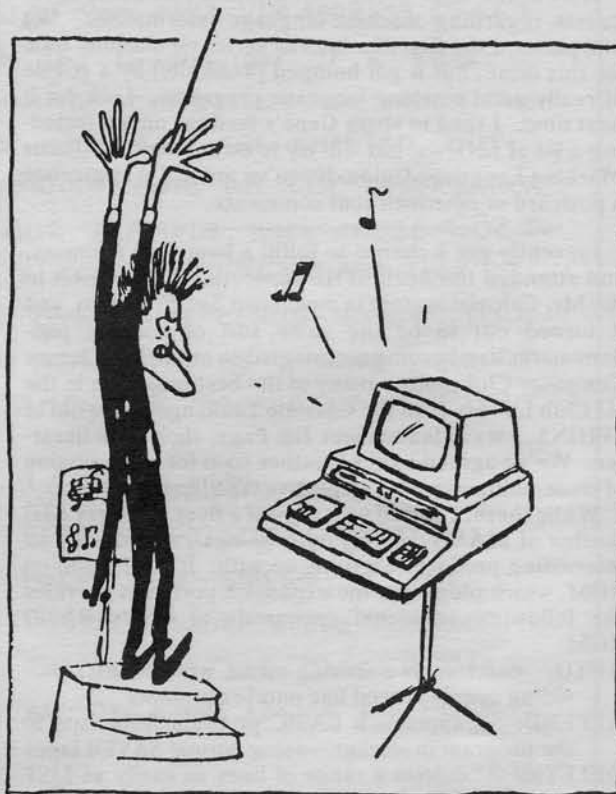
Once the BASIC program is loaded, it is no longer necessary to the operation of the test routine, although it is a little more convenient. Since the second cassette buffer is a protected location you can load and save and run BASIC programs while the test routine is tucked safely away. To test memory without the supporting BASIC program, simply POKE 1 and 2 with the low and high addresses of the starting point of the test and SYS 826. When control is transferred back to BASIC, PEEK locations 1 and 2 for the ending address—which will be either the top of memory in your system or the address of a bad byte. Pressing any key while the test is running will stop the test and transfer control back to BASIC.

```
5 REM MTEST JERRY A VELDER
8 FOR K=826 TO 877:READ D:POKE K,D:NEXTK
10 PRINT"ENTER START LOCATION IN DECIMAL"
15 PRINT"(DON'T START BELOW 1024)"
20 INPUT S
25 PRINT"WORKING..."
30 SH=INT(S/256)
40 SL=INT(S-(SH*256))
45 TI$="000000"
50 POKE 1,SL:POKE 2,SH
60 SYS826
65 EN=PEEK(2)*256+PEEK(1)
70 PRINT"TEST STOPPED AT ";EN
80 IF PEEK(2)=32 THEN PRINT "MEMORY OK"
85 T=TI/60
88 PRINT"EXECUTION TIME =";T;"SECONDS"
89 PRINT (EN-S)/T;"BYTES/SECOND"
90 END
100 DATA 160,0,162,0,161,1,133,70,152,129,1,193,1,208,36
105 DATA 200,208,246,165,70,129,1,24,165,1,105,1,133,1
110 DATA 165,2,105,0,133,2,169,0,32,228,255,208,9,165,2
115 DATA 201,132,240,3,76,62,3,96
READY.
```

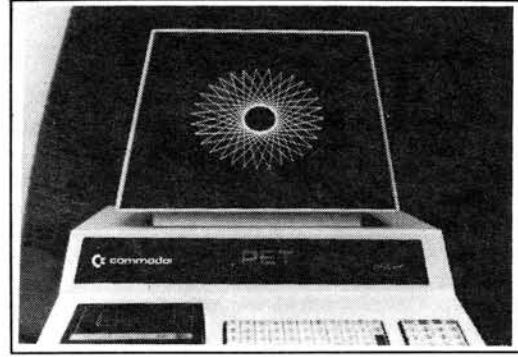
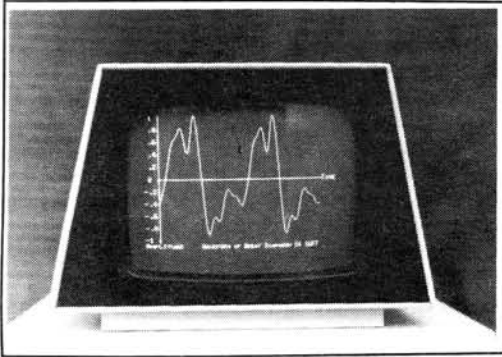
```
033A A0 00 LDY #0      ! clear Y reg(used as counter for test data)
033C A2 00 LDX #0      ! clear X reg
033E A1 01 LDA (1,X)    ! load acc with contents of 1st test byte
0340 85 46 STA 70       ! store in loc. 70 (dec)
0342 98 TYA            ! transfer test number to acc
0343 81 01 STA (1,X)    ! store in test loc
0345 C1 01 CMP (1,X)    ! compare with original
0347 D0 24 BNE $036D    ! if byte is bad, return to BASIC
0349 C8 INY            ! if test ok, increment test counter
034A D0 P6 BNE $0342    ! and try it for the next number
034C A5 46 LDA 70       ! when counter returns to 0, get original
034E 81 01 STA (1,X)    ! contents and restore them
0350 18 CLC            ! get ready to increment test address
0351 A5 01 LDA 1        !
0353 69 01 ADC #1       ! add 1 to low address
0355 85 01 STA 1        ! and store in loc 1
0357 A5 02 LDA 2        !
0359 69 00 ADC #0       ! add 0 to high address
035B 85 02 STA 2        ! and store in loc 2
035D A9 00 LDA #0       ! clear acc
035F 20 24 FF JSR $FFE4  ! see if any keys have been pressed
0362 D0 09 BNE $036D    ! if so, return to BASIC
0364 A5 02 LDA 2        ! load acc with high address
0366 C9 84 CMP #132     ! see if test has come to end of memory
0368 F0 03 BEQ $036D    ! if so, return to BASIC
036A 4C 3E 03 JMP $033E ! if not, test next address
036D 60 RTS            ! return to BASIC
```

Note: - Operands are decimal unless preceded by \$ (hex)
 - # denotes immediate addressing
 - (1,X) denotes indirect addressing, indexed by X
 - The CMP instruction in line 0366 is set for my system with 32K of memory + 1K screen RAM, all contiguous, but works OK on the basic 8K PET without modification.
 - Testing the video RAM (start at 32768) makes a nice, but brief, show.

Final note: Typing this out is a real pain. Anybody have any info on interfacing PET with a Singer 2303 Flexowriter?



REAL GRAPHICS FOR PET



How would you like to display some REAL graphics on your PET computer? Real graphics such as finely detailed mathematical curves or perspective drawings or computer art or proportionally spaced text or whatever directly on the PET screen. Graphics in which each individual dot in a huge 64,000 dot array 320 wide and 200 high is INDIVIDUALLY controllable. As a bonus, how would you like an extra 8K of memory when super graphics were not being used?

At last the famous MTU VISIBLE MEMORY available to KIM-1 users for well over a year is now available with an adaptor for the PET. The K-1007 PET-to-MTU adaptor plugs into the PET expansion connector and includes a cable and edge socket assembly which will accept our K-1008-PET Visible Memory board. Most all of the assembly can be hidden inside the PET if only the Visible Memory is used or the unit can be plugged into our K-1005 motherboard/card file for further expansion with other advanced design MTU boards. Use of the K-1007 does not tie up the PET's expansion port however.

Besides the bus adaptor, the K-1007 contains video processing circuitry to translate standard Visible Memory composite video into the form required by the PET monitor although composite VM video is still available for an external CCTV monitor. An on-board latch allows software switching between standard PET video and VM video. All necessary operating power is taken from the PET transformer but as most of our customers know, MTU boards require very little power for operation.

K-1008-PET \$243.00

K-1007 \$79.00

ALSO REAL 4 VOICE MUSIC

By now you have probably heard of the "DAC music system" for the KIM-1 that was pioneered by Hal Chamberlin in association with MTU. Up to 4 independent, simultaneous musical voices, each with an independent waveform, can be synthesized for true 4-part harmony. While software is the key to such performance, our K-1002-3C software package allows most of the potential of the technique to be realized on a standard 8K PET.

We have now redesigned the original DAC board specifically for the PET. The K-1002-2A operates from a single 5 volt power supply and includes an accurate 8 bit digital-to-analog converter, sharp 6 pole low-pass filter, and audio power amplifier usable with any kind of speaker. The board plugs directly onto the PET user port and second cassette port for power yet all signals feed-through to a second set of board edge fingers.

K-1002-2 \$50.00

K-1002-3C \$20.00

WANT MORE EXPANSION YET?

K-1005 5 Slot Card File \$75.00

K-1016 16K Low Power Memory \$340.00

K-1012 24K PROM, PROGRAMMER, I/O, COMM \$237.00

K-1020 Prototyping Board \$42.00

MICRO TECHNOLOGY UNLIMITED

841 Galaxy Way, P.O. Box 4596 • Manchester, NH 03108
603-627-1464

Son of 'DISABLING THE PET STOP KEY'

Jim Butterfield

1. Credit: adapted from an unknown source which later proved to be Richard Tobey of SPHINX.
2. New method: the STOP key and the clocks (TI and TIS) may be disabled with this simple POKE statement: POKE 537,136. They may be restored with POKE 537,133. This new method was discovered by Steve Punter of Toronto.
3. As previously noted, the Tobey method Must be disabled before I/O activity can take place (tape read or write, or IEEE-488 activity).
The new Punter method does not prevent I/O. However, it must be noted that any such activity (e.g., OPEN, CLOSE, PRINT#, SAVE, etc.) will restore normal operation of the STOP key, so that 537 must be POKEd again.
There does not seem to be any method of disabling the STOP key during I/O activities.

Ed. Note: Jim's original article "Disabling the PET Stop Key" appeared in Vol I issue 6 of the Notes. —O'B

FOUR-LINER DEP'T...

Richard Larson

Here is another fascinating four-liner your readers may enjoy. It selects a random character and fills a 25 x 26 square at the center of the screen using a spiral motion. Then it unwinds with a randomly selected reverse character.

As written, alphabetic characters are suppressed. To allow these characters, change the value of C in line 1 to read: $C = \text{RND}(1) * 128 - (Q = 2) * 128$. Also a clear screen could be squeezed into line 1.

Maybe I should add the warning that this program may be hypnotizing.

```
1 C=RND(1)*101+27-(Q=2)*128:IFP=0THENB=
33267:E=33760:P=B:R=1:W=-40:N=-1:Q=-2
```

```
2 S=-(Q=2):X=W:W=-N:N=-X:Q=-Q:X=R:R=L:
L=X:IFP=BTHENPOKE B,C
```

```
3 FORA=STOR:P=P+W:POKEP,C:NEXT:FORA=STO
R:P=P+N:POKEP,C:NEXT:IFP>EORP=BGOTO1
```

```
4 FORA=STOL:P=P-W:POKEP,C:NEXT:FORA=STO
L:P=P-N:POKEP,C:NEXT:R=R+Q:L=L+Q:GOTO3
```

B

PET User Group

Box 379

South Bound Brook, NJ 08880

FIRST CLASS

